Regression

www.biostat.wisc.edu/~dpage/cs760/
Goals for the lecture

you should understand the following concepts

- linear regression
- lasso
- least angles regression (LARS)
- RMSE, MAE, and R-square
- logistic regression
- logistic regression with lasso
Algorithms you should know

- closed-form solution for linear regression
- least angles regression (LARS) to solve lasso-penalized regression
- Logistic regression by gradient ascent or by LARS (if using lasso penalty)
Linear Regression

- Linear regression assumes that the relation between the expected value of dependent variable $Y$ and the value of independent variable(s) $X$, is linear.
Ordinary Least Square (OLS)

- For single variant, assume the data is given by
  \[ y_i = \alpha + \beta x_i + \varepsilon_i \]

where \( \varepsilon_i \) are Gaussian noises which are independent and have mean 0 and variance \( \sigma^2 \)
Ordinary Least Square (OLS)

- Goal: Minimize the objective function:
  \[
  error = \sum_i (h(x_i) - y_i)^2 \text{ or } \sum_i |h(x_i) - y_i|
  \]

- Solution:
  \[
  y = \alpha + \beta x + \varepsilon
  \]
  \[
  \hat{\beta} = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2}, \hat{\alpha} = \bar{y} - \hat{\beta} \bar{x}
  \]
Multivariate Linear Regression

- Write Matrix X and Y as:

\[ X = \begin{pmatrix} \vdots \\ \end{pmatrix} \quad \quad \quad Y = \begin{pmatrix} \vdots \\ \end{pmatrix} \]

- Solution:

\[ \hat{\beta} = (X^TX)^{-1}X^TY \]
LASSO (Penalized regression)

- Feature selection
  - Wrapper-based: Forward/Backward
  - Filter-based
  - LASSO: embed a penalty for more complex models in OBJ function.
    □ Define complex model.
LASSO: Penalty as a Constraint

Add penalty as a constraint to OBJ function:

Find $\hat{\alpha}$ and $\hat{\beta}$

To minimize $\sum_i (h(x_i) - y_i)^2$

Such that $\sum_j |\hat{\beta}_j|^{\text{error}} \leq s$ (s is a constant)
LASSO: Penalty as a Term in OBJ

Add penalty as a term to OBJ function to be minimized:

$$
\sum_i (h(x_i) - y_i)^2 + \lambda \sum_j |\hat{\beta}_j|
$$

(\lambda \text{ is penalty parameter})

No closed-form solution. Ordinary gradient ascent also does not work. Solve by quadratic programming (as in SVMs) or by modified gradient ascent called “least angles regression (LARS)” next…
Least Angles Regression (LARS)

- Set all $\beta_j = 0$.
- Find feature $x_j$ most correlated with response $y$, and move $\beta_j$ a small amount along the gradient and compute error $r$ (residual, or squared error for linear regression).
- Stop when some other predictor $x_k$ has as much correlation with $r$ as $x_j$ has.
- Continue (until can no longer reduce error) to move all currently nonzero coefficients jointly along the gradient in the same way, adding a new feature whenever it is as correlated with residual as current features. If a feature coefficient ever returns to zero, remove it.
Evaluation Metrics

- Root mean squared error (RMSE)
- Mean absolute error (MAE) – average error
- R-square (R-squared)
- Historically all were computed on training data, and possibly adjusted after, but really should cross-validate
**R-square(d)**

- **Formulation 1:**

\[ R^2 = 1 - \frac{\sum_i (y_i - h(x_i))^2}{\sum_i (y_i - \bar{y})^2} \]

- **Formulation 2:** square of Pearson correlation coefficient \( r \). Recall for \( x, y \):

\[ r = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\frac{1}{N} \sum_i (x_i - \bar{x})^2} \sqrt{\frac{1}{N} \sum_i (y_i - \bar{y})^2}} \]
Some Observations

• R-square of 0 means you have no model, R-square of 1 implies perfect model (loosely, explains all variation)
• These two formulations agree when performed on the training set
• The do not agree when we do cross-validation, in general, because mean of training set is different from mean of each fold
• Should do CV and use first formulation, but can be negative!
Logistic Regression: Motivation

• Linear regression was used to fit a linear model to the feature space.

• Requirement arose to do classification- make the number of interest the probability that a feature will take a particular value given other features

\[ P(Y = 1 \mid X) \]

• So, extend linear regression for classification
Theory of Logistic Regression

• To exhibit the relation between a dependent and an independent variable, we could use a step function. But it is not differentiable.

• We need a continuous and differentiable function: \textit{Logistic function}

\[
y = \frac{1}{1+e^{-cx}} = \frac{1}{1+e^{-wx}}
\]

\(wx \rightarrow \) a linear function of the feature vector \(x\)
Likelihood function for Logistic Regression

• Naïve Bayes Classification makes use of conditional information and computes full joint distribution
• Logistic regression instead computes conditional probability
  • operates on *Conditional Likelihood* function or *Conditional Log Likelihood*.
  • conditional Likelihood is the probability of the observed Y values conditioned on their corresponding X values
• Unlike linear regression, no closed-form solution, so we use standard *gradient ascent* to find the weight vector $w$ that maximizes the conditional likelihood
Learning Algorithm

• The conditional log likelihood is given by
  \[ l(w) = \ln(\prod_j P(y(j) \mid x(j) w)) \], where \( j \rightarrow j^{th} \) sample

• Need to find ‘\( w \)’ that maximizes the conditional log likelihood
  \[ \arg\max_w \ln(\prod_j P(y(j) \mid x(j) w)) \]

• Can use gradient ascent
  \[ w_i^{new} := w_i + \eta \frac{\partial l(w)}{\partial w_i} \], where \( \eta \rightarrow \text{learning rate parameter, } i \rightarrow i^{th} \) feature

• The derivative comes out to:
  \[ \frac{\partial l(w)}{\partial w_i} = \sum_j x_{i(j)} \left( y(j) - P(y(j) = 1 \mid x(j) w) \right) \]

• This gives us the gradient ascent rule:
  \[ w_i^{new} := w_i + \eta \sum_j x_{i(j)} \left( y(j) - P(y(j) = 1 \mid x(j) w) \right) \]

Error in estimate
Regularized Logistic Regression

- Logistic regression is prone to overfitting, especially when:
  - there are a large number of features or
  - when fit with high order polynomial features.
- Regularization helps combat overfitting by having a simpler model. It is used when we want to have:
  - less variation in the different weights or
  - smaller weights overall or
  - only a few non-zero weights (and thus features considered).
- Regularization is accomplished by adding a penalty term to the target function that is being optimized.
- Two types – $L_2$ and $L_1$ regularization.
$L_2$ regularization

• $L_2$ regularization uses 2-norm of the weight vector in the penalty term as shown:

$$\lambda \|w\|_2^2$$

i.e., $\lambda \sum_i w_i^2$ where ‘i’ represents the $i^{th}$ feature

• ‘$\lambda$’ is the regularization parameter used to control the weights. It governs how big the penalty is relative to fitting the data well.

• When the penalty term is added to the objective function, the gradient ascent algorithm’s update changes to:

$$w_i^{new} = w_i + \eta \sum_j x_{i(j)} \left( y_{(j)} - P \left( y_{(j)} = 1 \mid x_{(j)} w \right) \right) - \eta \lambda w_i$$

where, ‘$\eta$’ is the learning rate parameter
$L_1$ regularization

- L1 regularization uses 1-norm of the weight vector in the penalty term as shown:
  $$\lambda ||w||_1$$

  i.e., $\lambda \sum_i |w_i|$ where ‘i’ represents the $i^{th}$ feature

- Also called ‘Lasso’ penalty.

- Gradient ascent is no longer feasible since L1 norm is not differentiable.

- Can use Least Angle Regression (LARS).
Comments on regularization

• Logistic regression is prone to overfitting.
• Regularization helps combat overfitting by adding a penalty term to the target function being optimized.
• L1 regularization is often preferred since it produces sparse models. It can drive certain co-efficients(weights) to zero, performing feature selection in effect.
• L2 regularization drives towards smaller and simpler weight vectors but cannot perform feature selection like L1 regularization.
Comments on logistic regression

• Logistic Regression is a linear classifier
• In Bayesian classifier, features are independent given class -> assumption on P(X|Y)
• In Logistic Regression: Functional form is P(Y|X), there is no assumption on P(X|Y)
• Logistic Regression optimized by using conditional likelihood
  • no closed-form solution
  • concave -> global optimum with gradient ascent