Regularized, Polynomial, Logistic Regression

Pradeep Ravikumar

Co-instructor: Ziv Bar-Joseph

Machine Learning 10-701
Regression algorithms

Training data
\[ \{(X_i, Y_i)\}_{i=1}^{n} \]

Learning algorithm

Prediction rule
\[ \hat{f}_n \]
that predicts/estimates output \( Y \) given input \( X \)

Linear Regression
Regularized Linear Regression – Ridge regression, Lasso
Polynomial Regression
Gaussian Process Regression
...

hat{p}redicts/estimates output Y given input X
Recap: Linear Regression

\[ \hat{f}_n^L = \arg \min_{f \in \mathcal{F}_L} \frac{1}{n} \sum_{i=1}^{n} (f(X_i) - Y_i)^2 \]

\[ \mathcal{F}_L - \text{Class of Linear functions} \]

Uni-variate case:

\[ f(X) = \beta_1 + \beta_2 X \]

\[ \beta_1 - \text{intercept} \]

Multi-variate case:

\[ f(X) = f(X^{(1)}, \ldots, X^{(p)}) = \beta_1 X^{(1)} + \beta_2 X^{(2)} + \cdots + \beta_p X^{(p)} \]

\[ = X \beta \quad \text{where} \quad X = [X^{(1)} \ldots X^{(p)}], \quad \beta = [\beta_1 \ldots \beta_p]^T \]

Least Squares Estimator
Recap: Least Squares Estimator

$$\hat{f}_n^L = \arg \min_{f \in \mathcal{F}_L} \frac{1}{n} \sum_{i=1}^{n} (f(X_i) - Y_i)^2 \quad f(X_i) = X_i \beta$$

$$\hat{\beta} = \arg \min_{\beta} \frac{1}{n} \sum_{i=1}^{n} (X_i \beta - Y_i)^2 \quad \hat{f}_n^L(X) = X \hat{\beta}$$

$$= \arg \min_{\beta} \frac{1}{n} (A \beta - Y)^T (A \beta - Y)$$

$$A = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix} = \begin{bmatrix} X_1^{(1)} & \cdots & X_1^{(p)} \\ \vdots & \ddots & \vdots \\ X_n^{(1)} & \cdots & X_n^{(p)} \end{bmatrix} \quad Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix}$$
Recap: Least Square solution satisfies Normal Equations

\[(A^T A) \hat{\beta} = A^T Y\]

If \((A^T A)\) is invertible,

\[
\hat{\beta} = (A^T A)^{-1} A^T Y \quad \hat{f}_n^L(X) = X\hat{\beta}
\]

When is \((A^T A)\) invertible?
Recall: Full rank matrices are invertible. What is rank of \((A^T A)\)?

\[
\text{Rank}(A^T A) = \text{number of non-zero eigenvalues of } (A^T A) \\
\leq \min(n,p) \text{ since } A \text{ is } n \times p
\]

So, \(\text{rank}(A^T A) =: r \leq \min(n,p)\)
Not invertible if \(r < p\) (e.g. \(n < p\) i.e. high-dimensional setting)
Regularized Least Squares

What if \((A^T A)\) is not invertible?

r equations, p unknowns – underdetermined system of linear equations
many feasible solutions
Need to constrain solution further

e.g. bias solution to “small” values of \(\beta\) (small changes in input don’t translate to large changes in output)

\[
\hat{\beta}_{MAP} = \arg \min_\beta \sum_{i=1}^n (Y_i - X_i \beta)^2 + \lambda \|\beta\|_2^2
\]

\[
= \arg \min_\beta (A\beta - Y)^T (A\beta - Y) + \lambda \|\beta\|_2^2
\]

\[
\hat{\beta}_{MAP} = (A^T A + \lambda I)^{-1} A^T Y
\]

Is \((A^T A + \lambda I)\) invertible?
Understanding regularized Least Squares

\[
\min_{\beta} (A\beta - Y)^T (A\beta - Y) + \lambda \text{pen}(\beta) = \min_{\beta} J(\beta) + \lambda \text{pen}(\beta)
\]

Ridge Regression:
\[
\text{pen}(\beta) = \|\beta\|_2^2
\]

\(\beta\)s with constant \(J(\beta)\)
(level sets of \(J(\beta)\))

\(\beta\)s with constant l2 norm
(level sets of \(\text{pen}(\beta)\))

Unregularized Least Squares solution
Regularized Least Squares

What if \((A^T A)\) is not invertible?

\(r\) equations, \(p\) unknowns – underdetermined system of linear equations
many feasible solutions
Need to constrain solution further

e.g. bias solution to “small” values of \(b\) (small changes in input don’t translate to large changes in output)

\[
\hat{\beta}_{\text{MAP}} = \arg\min_{\beta} \sum_{i=1}^{n} (Y_i - X_i \beta)^2 + \lambda \|\beta\|_2^2
\]

Ridge Regression
(l2 penalty)

\[
\hat{\beta}_{\text{MAP}} = \arg\min_{\beta} \sum_{i=1}^{n} (Y_i - X_i \beta)^2 + \lambda \|\beta\|_1
\]

Lasso
(l1 penalty)

Many parameter values can be zero – many inputs are irrelevant to prediction in high-dimensional settings
Regularized Least Squares

What if \((A^TA)\) is not invertible?

\(r\) equations, \(p\) unknowns – underdetermined system of linear equations
many feasible solutions
Need to constrain solution further

\(\text{e.g. bias solution to “small” values of } \beta\) (small changes in input don’t translate to large changes in output)

\[
\hat{\beta}_{\text{MAP}} = \arg \min_{\beta} \sum_{i=1}^{n} (Y_i - X_i\beta)^2 + \lambda \|\beta\|_2^2 \quad \text{Ridge Regression (l2 penalty)}
\]

\[
\hat{\beta}_{\text{MAP}} = \arg \min_{\beta} \sum_{i=1}^{n} (Y_i - X_i\beta)^2 + \lambda \|\beta\|_1 \quad \text{Lasso (l1 penalty)}
\]

No closed form solution, but can optimize using sub-gradient descent (packages available)
Ridge Regression vs Lasso

$$\min_{\beta} (A\beta - Y)^T(A\beta - Y) + \lambda \text{pen}(\beta) = \min_{\beta} J(\beta) + \lambda \text{pen}(\beta)$$

Ridge Regression:
$$\text{pen}(\beta) = ||\beta||_2^2$$

Lasso:
$$\text{pen}(\beta) = ||\beta||_1$$

Ideally l0 penalty, but optimization becomes non-convex

Lasso (l1 penalty) results in sparse solutions – vector with more zero coordinates
Good for high-dimensional problems – don’t have to store all coordinates, interpretable solution!
Lasso vs Ridge

Lasso Coefficients

Ridge Coefficients
Regularized Least Squares – connection to MLE and MAP (Model-based approaches)
Least Squares and M(C)LE

Intuition: Signal plus (zero-mean) Noise model

\[ Y = f^*(X) + \epsilon = X\beta^* + \epsilon \]

\( \epsilon \sim \mathcal{N}(0, \sigma^2 I) \)
\( Y \sim \mathcal{N}(X\beta^*, \sigma^2 I) \)

\[ \hat{\beta}_{\text{MLE}} = \arg \max_{\beta} \log p(\{Y_i\}_{i=1}^n | \beta, \sigma^2, \{X_i\}_{i=1}^n) \]

Conditional log likelihood

\[ = \arg \min_{\beta} \sum_{i=1}^n (X_i\beta - Y_i)^2 = \hat{\beta} \]

Least Square Estimate is same as Maximum Conditional Likelihood Estimate under a Gaussian model!
Regularized Least Squares and M(C)AP

What if \((A^T A)\) is not invertible?

\[
\hat{\beta}_{\text{MAP}} = \arg \max_{\beta} \log p(\{Y_i\}_{i=1}^{n}|\beta, \sigma^2, \{X_i\}_{i=1}^{n}) + \log p(\beta)
\]

Conditional log likelihood

log prior

I) Gaussian Prior

\(\beta \sim \mathcal{N}(0, \tau^2 I)\)

\(p(\beta) \propto e^{-\beta^T \beta / 2\tau^2}\)

\[
\hat{\beta}_{\text{MAP}} = \arg \min_{\beta} \sum_{i=1}^{n} (Y_i - X_i \beta)^2 + \lambda \|\beta\|_2^2
\]

constant(\(\sigma^2, \tau^2\))

\[
\hat{\beta}_{\text{MAP}} = (A^T A + \lambda I)^{-1} A^T Y
\]

Ridge Regression
Regularized Least Squares and M(C)AP

What if \((A^T A)\) is not invertible?

\[
\hat{\beta}_{\text{MAP}} = \arg \max_{\beta} \log p(\{Y_i\}_{i=1}^n | \beta, \sigma^2, \{X_i\}_{i=1}^n) + \log p(\beta)
\]

I) Gaussian Prior

\[
\beta \sim \mathcal{N}(0, \tau^2I) \quad p(\beta) \propto e^{-\beta^T \beta / 2\tau^2}
\]

\[
\hat{\beta}_{\text{MAP}} = \arg \min_{\beta} \sum_{i=1}^n (Y_i - X_i \beta)^2 + \lambda \|\beta\|_2^2
\]

Ridge Regression

Prior belief that \(\beta\) is Gaussian with zero-mean biases solution to “small” \(\beta\)
Regularized Least Squares and M(C)AP

What if \((A^TA)\) is not invertible?

\[
\hat{\beta}_{\text{MAP}} = \arg \max_{\beta} \log p(\{Y_i\}_{i=1}^n | \beta, \sigma^2, \{X_i\}_{i=1}^n) + \log p(\beta)
\]

II) Laplace Prior

\[\beta_i \overset{iid}{\sim} \text{Laplace}(0, t)\]

\[p(\beta_i) \propto e^{-|\beta_i|/t}\]

\[
\hat{\beta}_{\text{MAP}} = \arg \min_{\beta} \sum_{i=1}^n (Y_i - X_i\beta)^2 + \lambda \|\beta\|_1
\]

Prior belief that \(\beta\) is Laplace with zero-mean biases solution to “sparse” \(\beta\)

Lasso
Beyond Linear Regression

Polynomial regression
Regression with nonlinear features
Polynomial Regression

Univariate (1-dim) case:

\[ f(X) = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_m X^m = X\beta \]

where \( X = [1 \ X \ X^2 \ldots X^m] \quad \beta = [\beta_1 \ldots \beta_m]^T \)

\[
\hat{\beta} = (A^T A)^{-1} A^T Y \quad \text{or} \quad (A^T A + \lambda I)^{-1} A^T Y \\
\hat{f}_n(X) = X\hat{\beta}
\]

where \( A = \begin{bmatrix} 1 & X_1 & X_1^2 & \ldots & X_1^m \\ \vdots & \vdots & \ddots & \vdots \\ 1 & X_n & X_n^2 & \ldots & X_n^m \end{bmatrix} \)

Multivariate (p-dim) case:

\[ f(X) = \beta_0 + \beta_1 X^{(1)} + \beta_2 X^{(2)} + \cdots + \beta_p X^{(p)} \]

\[ + \sum_{i=1}^{p} \sum_{j=1}^{p} \beta_{ij} X^{(i)} X^{(j)} + \sum_{i=1}^{p} \sum_{j=1}^{p} \sum_{k=1}^{p} X^{(i)} X^{(j)} X^{(k)} \]

\[ + \ldots \text{terms up to degree } m \]
Polynomial Regression

Polynomial of order $k$, equivalently of degree up to $k-1$

$k=1$

$k=2$

$k=3$

$k=7$

What is the right order?
Bias – Variance Tradeoff

Large bias, Small variance – poor approximation but robust/stable

Small bias, Large variance – good approximation but unstable

3 Independent training datasets
Later in the course, we will show that

\[ E[(f(X) - f^*(X))^2] = \text{Bias}^2 + \text{Variance} \]

\[ \text{Bias} = E[f(X)] - f^*(X) \quad \text{..... How far is the model from “true function”} \]

\[ \text{Variance} = E[(f(X) - E[f(X)])^2] \quad \text{..... How variable/stable is the model} \]
Effect of Model Complexity

- Test error
- Variance
- Bias

Complexity of $F$
Effect of Model Complexity

![Graph showing the relationship between model complexity and test error, training error, and variance. The graph illustrates that as model complexity decreases, test error and training error both increase, indicating an increase in variance.]
Regression with basis functions

\[ f(X) = \sum_{j=0}^{m} \beta_j \phi_j(X) \]

Basis coefficients \( \beta_j \)

Basis functions (Linear combinations yield meaningful spaces)

Polynomial Basis

\[ \phi_0(X) \]
\[ \phi_1(X) \]
\[ \phi_2(X) \]
\[ \vdots \]

Fourier Basis

\[ \phi_0(X) \]
\[ \phi_1(X) \]
\[ \phi_2(X) \]
\[ \vdots \]

Good representation for periodic functions

Wavelet Basis

\[ \phi_0(X) \]
\[ \phi_1(X) \]
\[ \phi_2(X) \]
\[ \vdots \]

Good representation for local functions
Regression with nonlinear features

\[ f(X) = \sum_{j=0}^{m} \beta_j X^j = \sum_{j=0}^{m} \beta_j \phi_j(X) \]

- Weight of each feature
- Nonlinear features

In general, use any nonlinear features

- e.g. \( e^X \), \( \log X \), \( 1/X \), \( \sin(X) \), ...

\[
\hat{\beta} = (A^T A)^{-1} A^T Y \\
\text{or} \quad (A^T A + \lambda I)^{-1} A^T Y
\]

\[ \hat{f}_n(X) = X \hat{\beta} \]

\[ A = \begin{bmatrix}
\phi_0(X_1) & \phi_1(X_1) & \ldots & \phi_m(X_1) \\
\vdots & \ddots & \ddots & \vdots \\
\phi_0(X_n) & \phi_1(X_n) & \ldots & \phi_m(X_n)
\end{bmatrix} \]

\[ X = \begin{bmatrix}
\phi_0(X) & \phi_1(X) & \ldots & \phi_m(X)
\end{bmatrix} \]
Regression to Classification

Regression

X = Brain Scan

Y = Age of a subject

Classification

X = Cell Image

Anemic cell
Healthy cell

Y = Diagnosis

Can we predict the “probability” of class label being Anemic or Healthy – a real number – using regression methods?

But output (probability) needs to be in [0,1]
Logistic Regression

Assumes the following functional form for $P(Y|X)$:

$$P(Y = 0|X) = \frac{1}{1 + \exp(w_0 + \sum_i w_i X_i)}$$

Logistic function applied to a linear function of the data

Logistic function (or Sigmoid):

$$\frac{1}{1 + \exp(-z)}$$

Features can be discrete or continuous!
Logistic Regression is a Linear Classifier!

Assumes the following functional form for $P(Y|X)$:

$$ P(Y = 0|X) = \frac{1}{1 + \exp(w_0 + \sum_i w_i X_i)} $$

Decision boundary: Note - Labels are 0,1

$$ P(Y = 0|X) \geq P(Y = 1|X) $$

$$ w_0 + \sum_i w_i X_i \geq 0 $$

(Linear Decision Boundary)
Logistic Regression is a Linear Classifier!

Assumes the following functional form for $P(Y|X)$:

$$P(Y = 0|X) = \frac{1}{1 + \exp(w_0 + \sum_i w_i X_i)}$$

$$\Rightarrow P(Y = 1|X) = \frac{\exp(w_0 + \sum_i w_i X_i)}{1 + \exp(w_0 + \sum_i w_i X_i)}$$

$$\Rightarrow \frac{P(Y = 1|X)}{P(Y = 0|X)} = \exp(w_0 + \sum_i w_i X_i) \stackrel{!}{\geq} 1$$

$$\Rightarrow w_0 + \sum_i w_i X_i \stackrel{!}{\geq} 0$$
Training Logistic Regression

How to learn the parameters $w_0$, $w_1$, ... $w_d$? (d features)

Training Data  $\{(X^{(j)}, Y^{(j)})\}_{j=1}^{n}$  $X^{(j)} = (X_1^{(j)}, \ldots, X_d^{(j)})$

Maximum Likelihood Estimates

$$\hat{w}_{MLE} = \arg \max_{w} \prod_{j=1}^{n} P(X^{(j)}, Y^{(j)} \mid w)$$

But there is a problem ...  

Don’t have a model for $P(X)$ or $P(X \mid Y)$ – only for $P(Y \mid X)$
Training Logistic Regression

How to learn the parameters $w_0$, $w_1$, ... $w_d$? (d features)

Training Data

$$\{(X^{(j)}, Y^{(j)})\}_{j=1}^n$$

$$X^{(j)} = (X_1^{(j)}, \ldots, X_d^{(j)})$$

Maximum (Conditional) Likelihood Estimates

$$\hat{w}_{MCLE} = \arg\max_w \prod_{j=1}^n P(Y^{(j)} \mid X^{(j)}, w)$$

Discriminative philosophy – Don’t waste effort learning $P(X)$, focus on $P(Y \mid X)$ – that’s all that matters for classification!
Expressing Conditional log Likelihood

\[ P(Y = 0|X, w) = \frac{1}{1 + \exp(w_0 + \sum_i w_i X_i)} \]

\[ P(Y = 1|X, w) = \frac{\exp(w_0 + \sum_i w_i X_i)}{1 + \exp(w_0 + \sum_i w_i X_i)} \]

\[ l(w) \equiv \ln \prod_j P(y^j| x^j, w) \]

\[ = \sum_j \left[ y^j (w_0 + \sum_i w_i x_i^j) - \ln(1 + \exp(w_0 + \sum_i w_i x_i^j)) \right] \]

**Bad news:** no closed-form solution to maximize \( l(w) \)

**Good news:** \( l(w) \) is concave function of \( w \)
concave functions easy to maximize
A function \( l(w) \) is called **concave** if the line joining two points \( l(w_1), l(w_2) \) on the function does not go above the function on the interval \([w_1, w_2]\). 

(Strictly) Concave functions have a unique maximum!
Optimizing concave function

- Conditional likelihood for Logistic Regression is concave
- Maximum of a concave function can be reached by

**Gradient Ascent Algorithm**

Initialize: Pick \( w \) at random

**Gradient:**

\[
\nabla_w l(w) = \left[ \frac{\partial l(w)}{\partial w_0}, \ldots, \frac{\partial l(w)}{\partial w_k} \right]
\]

**Update rule:**

\[
\Delta w = \eta \nabla_w l(w)
\]

\[
w_i(t+1) \leftarrow w_i(t) + \eta \frac{\partial l(w)}{\partial w_i} \bigg|_t
\]
Gradient Ascent for Logistic Regression

Gradient ascent rule for $w_0$:

$$w_0^{(t+1)} \leftarrow w_0^{(t)} + \eta \left. \frac{\partial l(w)}{\partial w_0} \right|_t$$

$$l(w) = \sum_j \left[ y^j (w_0 + \sum_i^d w_i x_i^j) - \ln(1 + \exp(w_0 + \sum_i^d w_i x_i^j)) \right]$$

$$\frac{\partial l(w)}{\partial w_0} = \sum_j \left[ y^j - \frac{1}{1 + \exp(w_0 + \sum_i^d w_i x_i^j)} \cdot \exp(w_0 + \sum_i^d w_i x_i^j) \right]$$

$$w_0^{(t+1)} \leftarrow w_0^{(t)} + \eta \sum_j [y^j - \hat{P}(Y^j = 1 \mid x^j, w^{(t)})]$$
Gradient Ascent for Logistic Regression

Gradient ascent algorithm: iterate until change < $\varepsilon$

$$w_0^{(t+1)} \leftarrow w_0^{(t)} + \eta \sum_j [y^j - \hat{P}(Y^j = 1 \mid x^j, w^{(t)})]$$

For $i=1,...,d$, 

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \sum_j x_i^j [y^j - \hat{P}(Y^j = 1 \mid x^j, w^{(t)})]$$

repeat

• Gradient ascent is simplest of optimization approaches
  – e.g., Newton method, Conjugate gradient ascent, IRLS (see Bishop 4.3.3)
That’s all M(C)LE. How about M(C)AP?

\[ p(w \mid Y, X) \propto P(Y \mid X, w)p(w) \]

- Define priors on \( w \)
  - Common assumption: Normal distribution, zero mean, identity covariance
  - “Pushes” parameters towards zero

\[ p(w) = \prod_i \frac{1}{\kappa \sqrt{2\pi}} \frac{-w_i^2}{e^{2\kappa^2}} \]

Zero-mean Gaussian prior

- M(C)AP estimate

\[
w^* = \arg \max_w \ln \left[ p(w) \prod_{j=1}^n P(y^j \mid x^j, w) \right]
\]

\[
w^* = \arg \max_w \sum_{j=1}^n \ln P(y^j \mid x^j, w) - \sum_{i=1}^d \frac{w_i^2}{2\kappa^2}
\]

Still concave objective!

Penalizes large weights
M(C)AP – Gradient

- Gradient

\[
\frac{\partial}{\partial w_i} \ln p(w) + \frac{\partial}{\partial w_i} \ln \left[ \prod_{j=1}^{n} P(y_j | x_j, w) \right]
\]

- Zero-mean Gaussian prior

\[
p(w) = \prod_i \frac{1}{\kappa \sqrt{2\pi}} e^{-\frac{w_i^2}{2\kappa^2}}
\]

- Same as before

- Extra term Penalizes large weights

\[
\propto \frac{-w_i}{\kappa^2}
\]
M(C)LE vs. M(C)AP

• Maximum conditional likelihood estimate

\[ w^* = \arg \max_w \ln \left( \prod_{j=1}^{n} P(y^j \mid x^j, w) \right) \]

\[ w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \sum_j x_i^j [y^j - P(Y = 1 \mid x^j, w^{(t)})] \]

• Maximum conditional a posteriori estimate

\[ w^* = \arg \max_w \ln \left( p(w) \prod_{j=1}^{n} P(y^j \mid x^j, w) \right) \]

\[ w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \left\{ -\frac{1}{\kappa^2} w_i^{(t)} + \sum_j x_i^j [y^j - P(Y = 1 \mid x^j, w^{(t)})] \right\} \]
Logistic Regression for more than 2 classes

• Logistic regression in more general case, where \( Y \in \{y_1, \ldots, y_K\} \)

  for \( k < K \)

\[
P(Y = y_k | X) = \frac{\exp(w_{k0} + \sum_{i=1}^{d} w_{ki}X_i)}{1 + \sum_{j=1}^{K-1} \exp(w_{j0} + \sum_{i=1}^{d} w_{ji}X_i)}
\]

  for \( k = K \) (normalization, so no weights for this class)

\[
P(Y = y_K | X) = \frac{1}{1 + \sum_{j=1}^{K-1} \exp(w_{j0} + \sum_{i=1}^{d} w_{ji}X_i)}
\]

Predict \( f^*(x) = \arg \max_{Y=y} P(Y = y | X = x) \)

Is the decision boundary still linear?