Regression

CPS 271
Ron Parr

Supervised Learning

- Given: Training Set
- Goal: Good performance on test set
- Assumptions:
  - Training samples are independently drawn, and identically distributed (IID)
  - Test set is from same distribution as training set

Regression Specifics

- Datum $i$ has feature vector: $x^{(i)}$
- Has real valued target: $y^{(i)}$
- Space of concepts $H =$ linear combinations of feature vectors: $h(x) = \theta^T x$
- Learning objective: Search to find "best" $\theta$
  - (This is standard "data fitting" that most people learn in some form or another.)

Linearity of Regression

- Regression typically considered a linear method, but...
  - Features not necessarily linear
  - Features not necessarily linear
  - Features not necessarily linear
  - Features not necessarily linear
  - and, BTW, features not necessarily linear

Regression Examples

- Predicting housing price from:
  - House size, lot size, rooms, neighborhood*, etc.
- Predicting weight from:
  - Sex, height, ethnicity, etc.
- Predicting life expectancy increase from:
  - Medication, disease state, etc.
- Predicting crop yield from:
  - Precipitation, fertilizer, temperature, etc.

What is "best"?

- No obvious answer to this question
- Three compatible answers:
  - Minimize squared error on training set
  - Maximize likelihood of the data (under certain assumptions)
  - Project data into "closest" approximation
- Other answers possible
Minimizing Squared Training Set Error

- Why is this good?
- How could this be bad?
- Minimize:
  \[ J(\theta) = \sum_{i=1}^{n} (\theta \cdot x^{(i)} - y^{(i)})^2 \]

Maximizing Likelihood of Data

- Assume:
  - True model is in \( H \)
  - Data have Gaussian noise
- Actually might want:
  \[
  \arg \max_{\theta} P(H \mid X) = \frac{P(X \mid H)P(H)}{P(X)}
  \]
- Is maximizing \( P(X \mid H) \) a good surrogate? (maximizing over \( \theta \))

Maximizing \( P(X \mid H) \)

- Assume: \( y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)} \)
- Where: \( P(\epsilon^{(i)}) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{\epsilon^{(i)}^2}{2\sigma^2}\right) \)
  (Gaussian distribution w/ mean 0, standard deviation \( \sigma \))
- Therefore:
  \[
  P(y^{(i)} \mid x^{(i)}, \theta) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)
  \]

Maximization Continued

- Maximizing over entire data set:
  \[
  \prod_{i} P(y^{(i)} \mid x^{(i)}, \theta) = \prod_{i} \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)
  \]
- Maximizing equivalent log formulation:
  (ignoring constants)
  \[
  \sum_{i} (y^{(i)} - \theta^T x^{(i)})^2
  \]
  or minimizing:
  \[
  \sum_{i} (y^{(i)} - \theta^T x^{(i)})^2
  \]
  Look familiar?

Checkpoint

- So far we have considered:
  - Minimizing squared error on training set
  - Maximizing Likelihood of training set
    (given model, and some assumptions)
- Different approaches w/same objective!

Design Matrix

- We call \( A \) the design matrix
- Columns of \( A \) are features
- Rows of \( A \) are data
- \( a_{ij} = \text{Feature } j \text{ of training instance } i \)
**Geometric Interpretation**

- \( Y = (Y^{(1)}, \ldots, Y^{(n)}) \) = point in \( n \)-space
- \( A \theta = H = \) column space of features
- \( A \theta = \) subspace of \( \mathbb{R}^n \) occupied by \( H \)
- Goal: Find "closest" point in \( H \) to \( Y \)
- Suppose closeness = Euclidean distance

**Minimizing Euclidean Distance**

- Minimize: \( \| Y - A \theta \|_2 \)
- For \( n \) data points:
  \[
  \sum_{i=1}^{N} (y^{(i)} - x^{(i)} \theta)^2
  \]
- Equivalent to minimizing:
  \[
  \sum_{i=1}^{N} (y^{(i)} - x^{(i)} \theta)^2
  \]

**Another Geometric Interpretation**

![Diagram](image)

**Checkpoint**

- Three different ways to pick \( H \)
  - Minimize squared error on training set
  - Maximize likelihood of training set
  - Distance minimizing projection into \( H \)
- All lead to same optimization problem!
  \[
  \arg\min_{\theta} J(\theta) = \sum_{i=1}^{N} (\theta \cdot x^{(i)} - y^{(i)})^2
  \]

**Solving the Optimization Problem**

- Nota bene: Good to keep optimization problem and optimization technique separate in your mind
- Some optimization approaches:
  - Gradient descent
  - Direct Minimization derived from
    - Calculus
    - Geometric constraints

**Minimizing J by Gradient Descent**

![Diagram](image)

(Adapted from Luke Golub's Slides)
Gradient Descent Issues

- For this particular problem:
  - Global minimum exists
  - Convergence guaranteed if done in "batch"
- In general:
  - Local optimum only
  - Batch mode more stable
  - Incremental possible
    - Can oscillate
    - Use decreasing step size (Rabin-Monro) to stabilize

Direct Solution

- Geometric Approach (Strang)
- Let \( A \) be the design matrix
- Require orthogonality:
  \[
  \forall z : (Az - Y) = 0
  \]
  Any vector in \( H \)  
  Line from \( Y \) to solution
  \[
  \forall z : z^T [A^T A \theta - A^T Y] = 0
  \]

Direct Solution Continued

- When is this true: \( \forall z: z^T [A^T A \theta - A^T Y] = 0 \)
- When:
  \[
  A^T A \theta - A^T Y = 0
  \theta = (A^T A)^{-1} A^T Y
  \]

  When does the inverse exist?

  Columns of \( A \) must be independent.

What about other criteria?

- How about minimizing worse case loss?
  \[
  \min_{\theta} \max \left( \theta \cdot x^{(i)} - y^{(i)} \right)
  \]
- Solve by linear program…