Day 2: Overfitting, regularization

Introduction to Machine Learning Summer School
June 18, 2018 - June 29, 2018, Chicago

Instructor: Suriya Gunasekar, TTI Chicago

19 June 2018
Review

• Yesterday
  o Supervised learning
  o Linear regression - polynomial curve fitting
  o Empirical risk minimization, evaluation

• Today
  o Overfitting
  o Model selection
  o Regularization
  o Gradient descent

• Schedule:
  9:00am-10:25am  Lecture 2.a: Overfitting, model selection
  10:35am-noon    Lecture 2.b: Regularization, gradient descent
  noon-1:00pm      Lunch
  1:00pm-3:30pm    Programming
  3:30pm-5:00pm    Invited Talk - Mathew Walter
Overfitting
Dataset size and linear regression

• Recall linear regression
  o Input $x \in \mathbb{R}^d$, output $y \in \mathbb{R}$, training data $S = \{(x^{(i)}, y^{(i)}): i = 1, 2, \ldots, N\}$
  o Estimate $\mathbf{w} \in \mathbb{R}^d$ and bias $w_0 \in \mathbb{R}$ by minimizing training loss

$$\hat{\mathbf{w}}, \hat{w}_0 = \operatorname{argmin}_{\mathbf{w}, w_0} \sum_{i=1}^{N} (\mathbf{w} \cdot x^{(i)} + w_0 - y^{(i)})^2$$

• What happens when we only have a single data point (in 1D)?
  o Ill-posed problem: an infinite number of lines perfectly fit the data
Dataset size and linear regression

- Recall linear regression
  - Input $x \in \mathbb{R}^d$, output $y \in \mathbb{R}$, training data $S = \{(x^{(i)}, y^{(i)}): i = 1, 2, ..., N\}$
  - Estimate $\mathbf{w} \in \mathbb{R}^d$ and bias $w_0 \in \mathbb{R}$ by minimizing training loss
    $$\hat{\mathbf{w}}, \hat{w}_0 = \operatorname{argmin}_{\mathbf{w}, w_0} \sum_{i=1}^{N} (\mathbf{w} \cdot x^{(i)} + w_0 - y^{(i)})^2$$

- What happens when we only have a single data point (in 1D)?
  - Ill-posed problem: an infinite number of lines perfectly fit the data

- Two points in 1D?
- Two points in 2D?
  - the amount of data needed to obtain a meaningful estimate of a model is related to the number of parameters in the model (its complexity)
Linear regression - generalization

Consider 1D example
- \( S_{\text{train}} = \{(x^{(i)}, y^{(i)}): i = 1,2, \ldots, N\} \) where
  - \( x^{(i)} \sim \text{uniform}(-5,5) \)
  - \( y^{(i)} = w^*x^{(i)} + \epsilon^{(i)} \) for true \( w^* \) and noise \( \epsilon^{(i)} \sim \mathcal{N}(0,1) \)
- \( S_{\text{test}} \) similarly generated
  \[
  \hat{w} = \arg\min_w \sum_{i=1}^{N} (wx^{(i)} - y^{(i)})^2
  \]
- The training error increases with the size of training data?
Model complexity vs fit for fixed N

• Recall polynomial regression of degree $m$ in 1D

$$\hat{w} = \underset{w \in \mathbb{R}^{m+1}}{\text{argmin}} \sum_{i=1}^{N} (w_0 + w_1 \cdot x^{(i)} + w_2 \cdot x^{(i)^2} + \cdots + w_m \cdot x^{(i)^m} - y_t)^2$$

N=30
Overfitting with ERM

• For same amount of data, more complex models overfits more than simple model
  o Recall: higher degree → more number of parameters to fit

• What happens if we have more data?
Model complexity vs fit for fixed N

• Recall polynomial regression of degree $m$ in 1D

$$\hat{w} = \arg\min_{w \in \mathbb{R}^{m+1}} \sum_{i=1}^{N} \left( w_0 + w_1 \cdot x^{(i)} + w_2 \cdot x^{(i)^2} + \cdots + w_m \cdot x^{(i)^m} - y_t \right)^2$$

![Graph](image)

N=100
Overfitting with ERM

• For same amount of data, complex models overfit more than simple models
  o Recall: higher degree $\rightarrow$ more number of parameters to fit

• What happens if we have more data?
  o More complex models require more data to avoid overfitting
How to avoid overfitting?

• How to **detect** overfitting?

  ![Graph showing model, true function, and samples with Degree 15 and Train MSE = 0.005.]

• How to **avoid** overfitting?
  - Look at test error and pick $m=5$?
  - Split $S = S_{\text{train}} \cup S_{\text{val}} \cup S_{\text{test}}$
    - Use performance on $S_{\text{val}}$ as proxy for test error

![Graph showing training error and test error with degree $m$, $N=30$.]
Model selection

- \( S = S_{\text{train}} \cup S_{\text{val}} \cup S_{\text{test}} \)
- m model classes \( \{ \mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_m \} \)
  - Recall each \( \mathcal{H}_r \) is a set of candidate functions mapping \( x \to y \)
  - e.g., \( \mathcal{H}_r = \{ x \to w_0 + w_1 x + w_2 x^2 + \cdots + w_r x^r \} \)
- Minimize training loss \( L_{S_{\text{train}}} \) on \( S_{\text{train}} \) to pick best \( \hat{f}_r \in \mathcal{H}_r \)
  - e.g., \( \hat{f}_r(x) = \hat{w}_0 + \hat{w}_1 x + \hat{w}_2 x^2 + \cdots \hat{w}_r x^r \) where \( \hat{w}_0, \hat{w}_1, \hat{w}_2, \ldots, \hat{w}_r \)

\[
= \arg\min_{w_0, \ldots, w_r} \sum_{(x(i), y(i)) \in S_{\text{train}}} \left( w_0 + w_1 x^{(i)} + w_2 x^{(i)^2} + \cdots + w_r x^{(i)^r} - y^{(i)} \right)^2
\]

- Compute validation loss \( L_{S_{\text{val}}} (\hat{f}_r) \) on \( S_{\text{val}} \) for each \( \{ \hat{f}_1, \hat{f}_2, \ldots, \hat{f}_m \} \)
- Pick \( \hat{f}^* = \min \{ L_{S_{\text{val}}} (\hat{f}_1), L_{S_{\text{val}}} (\hat{f}_2), \ldots, L_{S_{\text{val}}} (\hat{f}_m) \} = \min_r L_{S_{\text{val}}} (\hat{f}_r) \)
- Evaluate test loss \( L_{S_{\text{test}}} (\hat{f}^*) \)
Model selection

• Can we overfit to validation data?
  ○ How much data to keep aside for validation?
• What if we don’t have enough data?
Cross validation

Split \( S = S_1 \cup S_2 \cup \ldots \cup S_K \cup S_{test} \)

<table>
<thead>
<tr>
<th></th>
<th>( S_1 )</th>
<th>( S_2 )</th>
<th>( S_3 )</th>
<th>( S_4 )</th>
<th>( S_5 )</th>
<th>( S_{test} )</th>
</tr>
</thead>
</table>

- \( m \) model classes \( \{\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_m\} \)
- For each \( k \):
  - **Training loss** \( L_{S_{train}^k} \) is loss on \( S_{train}^k = S_1 \cup S_2 \ldots \cup S_{k-1} \cup S_{k+1} \ldots S_K \)
  - Let best \( \hat{f}_r^{(k)} \in \mathcal{H}_r \) by \( \hat{f}_r^{(k)} = \arg\min_{f \in \mathcal{H}_r} L_{S_{train}^k} (f) \)
  - Compute **validation loss** \( L_{S_k} (\hat{f}_r^{(k)}) \) on \( S_k \) for each \( r \)

- **Pick model based on average validation loss** \( \hat{r}^* = \arg\min_r \sum_{k=1}^K L_{S_k} (\hat{f}_r^{(k)}) \)
  - \( \mathcal{H}_{\hat{r}^*} \) is the correct model class to use.
  - \( \hat{f}^* = \arg\min_{f \in \mathcal{H}_{\hat{r}^*}} L_{S_{train}^k \cup S_k} (f) \) or \( \hat{f}^* = \sum_k \hat{f}_r^{(k)} \) (if it makes sense)
  - Evaluate \( L_{S_{test}} (\hat{f}^*) \)

Illustration credit: Nati Srebro

Extreme case \( K = N \) (leave one out cross validation)