Introduction to Machine Learning

Generalization and Model Validation

Prof. Andreas Krause
Learning and Adaptive Systems (las.ethz.ch)
Recall: Least-squares linear regression optimization

[Legendre 1805, Gauss 1809]

- Given data set \( D = \{(x_1, y_1), \ldots, (x_n, y_n)\} \)

\[
\hat{w} = \arg \min_w \sum_{i=1}^{n} (y_i - w^T x_i)^2
\]

- Last lecture, discussed how to solve using closed form & gradient descent
## Supervised learning summary so far

<table>
<thead>
<tr>
<th>Representation/features</th>
<th>Linear hypotheses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model/objective:</td>
<td>Loss-function</td>
</tr>
<tr>
<td></td>
<td>Squared loss, $l_p$ loss</td>
</tr>
<tr>
<td>Method:</td>
<td>Exact solution, Gradient Descent</td>
</tr>
<tr>
<td>Evaluation metric:</td>
<td>Empirical risk = (mean) squared error</td>
</tr>
</tbody>
</table>
Recall: Important choices in regression

- What types of functions $f$ should we consider? Examples

- How should we measure goodness of fit?

![Graph showing linear and non-linear models]

$f(x)$

$X$
Fitting nonlinear functions

How about functions like this:
Linear regression for polynomials

We can fit non-linear functions via linear regression, using nonlinear features of our data (basis functions)

\[ f(x) = \sum_{i=1}^{d} w_i \phi_i(x) \]
Demo: Linear regression on polynomials
Underfitting

Overfitting
### Supervised learning summary so far

<table>
<thead>
<tr>
<th>Representation/features</th>
<th>Linear hypotheses, nonlinear hypotheses through feature transformations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model/objective:</td>
<td>Loss-function</td>
</tr>
<tr>
<td></td>
<td>Squared loss, $l_p$-loss</td>
</tr>
<tr>
<td>Method:</td>
<td>Exact solution, Gradient Descent</td>
</tr>
<tr>
<td>Evaluation metric:</td>
<td>Mean squared error</td>
</tr>
</tbody>
</table>
Model selection for linear regression with polynomials

How can we estimate this?

Best model

Error

Prediction error

Training error

Degree of polynomial
Interlude: A note on probability

You’ll need to know about basic concepts in probability:

- Random variables
- Expectations (Mean, Variance etc.)
- Independence (i.i.d. samples from a distribution, ...)
- ...

...
Example: Gaussian distribution

\[ \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left( -\frac{(x - \mu)^2}{2\sigma^2} \right) \]

- \( \sigma \) = Standard deviation
- \( \mu \) = mean
Example: Multivariate Gaussian

\[
\frac{1}{2\pi \sqrt{|\Sigma|}} \exp \left( -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right)
\]

\[\Sigma = \begin{pmatrix}
\sigma_1^2 & \sigma_{12} \\
\sigma_{21} & \sigma_2^2
\end{pmatrix} \quad \mu = \begin{pmatrix}
\mu_1 \\
\mu_2
\end{pmatrix}
\]

\[
\Sigma = \begin{pmatrix}
1 & 0 \\
0 & 1
\end{pmatrix}
\]

\[
\Sigma = \begin{pmatrix}
1 & 0.9 \\
0.9 & 1
\end{pmatrix}
\]
Interlude: Expectations

- Expected value of random variable X

- Expected value of some function of X

- Linearity of expectation
Achieving generalization

Fundamental assumption: Our data set is generated independently and identically distributed (iid) from some unknown distribution $P$

$$(x_i, y_i) \sim P(X, Y)$$

Our goal is to minimize the expected error (true risk) under $P$

$$R(w) = \int P(x, y) (y - w^T x)^2 dx dy$$

$$= \mathbb{E}_{x,y} [(y - w^T x)^2]$$
Side note on iid assumption

When is iid assumption invalid?
- Time series data
- Spatially correlated data
- Correlated noise
- ...

Often, can still use machine learning, but one has to be careful in interpreting results.

Most important: Choose train/test to assess the desired generalization
Estimating the generalization error

- Estimate the true risk by the empirical risk on a sample data set $D$

$$
\hat{R}_D(w) = \frac{1}{|D|} \sum_{(x,y) \in D} (y - w^T x)^2
$$

- Why might this work?

**Law of large numbers**

$\hat{R}_D(w) \rightarrow R(w)$ for any fixed $w$ almost surely as $|D| \rightarrow \infty$
What happens if we optimize on training data?

Suppose we are given training data $D$.

**Empirical Risk Minimization:**

$$\hat{w}_D = \arg\min_w \hat{R}_D(w)$$

Ideally, we wish to solve

$$w^* = \arg\min_w R(w)$$
Empirical vs true risk

Risk

Generalization error

$R$

$\hat{R}$

$n=100$
Empirical vs true risk

Risk

Generalization error

$R$

$\hat{R}$

$n=1000$
Empirical vs true risk

Risk

Generalization error

$R$, $\hat{R}$

$n=10000$
Outlook: Requirements for learning

For learning via empirical risk minimization to be successful, need uniform convergence:

$$\sup_w |R(w) - \hat{R}_D(w)| \to 0 \text{ as } |D| \to \infty$$

This is not implied by law of large numbers alone, but depends on model class (holds, e.g., for squared loss on data distributions with bounded support) ➔ Statistical learning theory
What can go wrong in ERM

\[ n = |D| \]
What can go wrong in ERM

Risk

$\hat{R}$

$R$

$n=1000$

$w$

$w^*$

$\hat{w}$
What can go wrong in ERM

Risk

\( R \)

\( \hat{R} \)

\( n = 10000 \)
Learning from finite data

- Law of large numbers / uniform convergence are asymptotic statements (with \( n \to \infty \))
- In practice one has finite amount of data.

What can go wrong?
Simple example

\[ \hat{w}_D = \arg\min_w \hat{R}_D(w) \quad \quad w^* = \arg\min_w R(w) \]
What if we evaluate performance on training data?

\[ \hat{w}_D = \arg \min_w \hat{R}_D(w) \quad \quad \quad w^* = \arg \min_w R(w) \]

- In general, it holds that

\[ \mathbb{E}_D \left[ \hat{R}_D(\hat{w}_D) \right] < \mathbb{E}_D \left[ R(\hat{w}_D) \right] \]

- Thus, we obtain an overly optimistic estimate!
More realistic evaluation?

- Want to avoid underestimating the prediction error
- **Idea**: Use separate test set from the same distribution $P$
- Obtain training and test data $D_{train}$ and $D_{test}$
- Optimize $w$ on training set

$$
\hat{w} = \arg\min_w \hat{R}_{train}(w)
$$

- Evaluate on test set

$$
\hat{R}_{test}(\hat{w}) = \frac{1}{|D_{test}|} \sum_{(x,y) \in D_{test}} (y - \hat{w}^T x)^2
$$

- Then:

$$
\mathbb{E}[\hat{R}_{test}(\hat{w})] = \mathbb{E}[R(\hat{w})]
$$
Why?
Evaluating predictive performance

- Training error (empirical risk) **systematically underestimates** true risk

\[ \mathbb{E}_D \left[ \hat{R}_D(\hat{\mathbf{w}}_D) \right] < \mathbb{E}_D \left[ R(\hat{\mathbf{w}}_D) \right] \]

- Using an **independent test set** avoids this bias

\[ \mathbb{E}_{D_{\text{train}}, D_{\text{test}}} \left[ \hat{R}_{D_{\text{test}}}(\hat{\mathbf{w}}_{D_{\text{train}}}) \right] = \mathbb{E}_{D_{\text{train}}} \left[ R(\hat{\mathbf{w}}_{D_{\text{train}}}) \right] \]
First attempt: Evaluation for model selection

- Obtain training and test data $D_{train}$ and $D_{test}$
- Fit each candidate model (e.g., degree $m$ of polynomial)

$$\hat{w}_m = \arg\min_{\text{w: degree}(\text{w}) \leq m} \hat{R}_{train}(\text{w})$$

- Pick one that does best on test set:

$$\hat{m} = \arg\min_{m} \hat{R}_{test}(\hat{w}_m)$$

- Do you see a problem?
Overfitting to test set

Test error is itself random! Variance usually increases for more complex models

- Optimizing for single test set creates bias