Statistical Learning Theory

Nan Ye

School of Mathematics and Physics
The University of Queensland

Fundamental Questions

Different views of learning

- H. Simon: Any process by which a system improves its performance.
- M. Minsky: Learning is making useful changes in our minds.
- R. Michalsky: Learning is constructing or modifying representations of what is being experienced.
- L. Valiant: Learning is the process of knowledge acquisition in the absence of explicit programming.

- We have already seen a number of learning algorithms, including OLS, kNN regression/classification, logistic regression.
- Is there a useful theoretical notion of learning that can be used to study machine learning algorithms?

Questions

Learning theory considers a few fundamental questions.

- identification: can we learn the correct model given enough data?
- generalization: can we use the learned model to make predictions on new data?
- sample complexity: how many examples do we need to learn a good model?
- computational complexity: how much time do we need to learn a good model?

We focus on the discussion on the concept of generalization in this course, and show how it can be used to provide a unified view on some algorithms.

Recall

I pick a coin with the probability of heads being θ. I flip it 100 times for you and you see a dataset D of 70 heads and 30 tails, can you learn θ?

Recall

I pick a coin with the probability of heads being θ. I flip it 100 times for you and you see a dataset D of 70 heads and 30 tails, can you learn θ?

Maximum likelihood estimation
The likelihood of θ is

$$P(D \mid \theta) = \theta^{70}(1-\theta)^{30}.$$

Learning θ amounts to maximizing the likelihood.

$$\begin{aligned}
\theta_{ml} &= \arg\max_{\theta} \; P(D \mid \theta) \\
&= \arg\max_{\theta} \; \ln P(D \mid \theta) \\
&= \arg\max_{\theta} \; (70 \ln \theta + 30 \ln(1-\theta)).
\end{aligned}$$

$$\theta_{ml} = \arg\max_{\theta} \left(70 \ln \theta + 30 \ln(1-\theta)\right).$$

Set derivative of log-likelihood to 0,

$$\frac{70}{\theta} - \frac{30}{1-\theta} = 0,$$

we have

$$\theta_{ml} = 70/(70+30).$$

Statistical Learning Theory

Statistical learning theory provides a statistical formulation of the concept of learning (it is in line with H. Simon's view).

Data

Training examples $z_1, \ldots, z_n$ are i.i.d. drawn from a *fixed* but *unknown* distribution $P(Z)$ on $\mathcal{Z}$.

e.g. outcomes of coin flips.

Hypothesis space $\mathcal{H}$

This is the set of possible models.

e.g. head probability $\theta \in [0, 1]$.

Loss function

- This defines how well a model performs on an example.
- $L(z, h)$ measures the penalty for hypothesis h on example z.

$$\textit{e.g. log-loss } L(z,\theta) = -\ln P(z \mid \theta) = \begin{cases} -\ln(\theta), & z = H, \\ -\ln(1-\theta), & z = T. \end{cases}$$

Empirical risk

- This measures how model performs on the training data.
- That is, the empirical risk of a model h is

$$R_n(h) = \frac{1}{n} \sum_i L(z_i, h).$$

Expected risk

- This measures how a model performs on all possible data.
- The expected risk of a model h is

$$R(h) = \mathbb{E}(L(Z, h)),$$

  where the expectation is wrt the unknown data distribution $P(Z)$.
- A best model is one that minimizes the expected risk,

"Mother" of learning algorithms

Empirical risk minimization (ERM) chooses the model with the best performance on the training data, that is, we choose

$$h_n = \arg\min_{h \in \mathcal{H}} R_n(h).$$

e.g. choose θ *to minimize* $-70 \ln \theta - 30 \ln(1 - \theta)$.

This provides a unified formulation for many machine learning problems, which differ in

- the data domain $\mathcal{Z}$,
- the choice of the hypothesis space $\mathcal{H}$, and
- the choice of loss function L.

Most algorithms that we see later can be seen as special cases of ERM.

Your Turn

Bob proposes a new regression algorithm, which he calls the least cubes algorithm, by replacing the quadratic loss $L((x, y), h) = (h(x) - y)^2$ in OLS by the cubic loss

$$L((x, y), h) = (h(x) - y)^3.$$

Will you use this algorithm to solve your regression problems? Why?

Classification

- Given $D = \{(x_1, y_1), \ldots, (x_n, y_n)\} \subseteq \mathcal{X} \times \mathcal{Y}$, find a classifier f that maps an input $x \in \mathcal{X}$ to a *class* $y \in \mathcal{Y}$.
- We usually use the 0/1 loss

$$L((x, y), h) = I(h(x) = y) = \begin{cases} 1, & h(x) \neq y, \\ 0, & h(x) = y. \end{cases}.$$

- ERM chooses the classifier with minimum classification error

$$\min_{h \in \mathcal{H}} \frac{1}{n} \sum_i I(h(x_i) \neq y_i).$$

Bayes Optimal Classifier

The expected 0/1 loss is minimized by the classifier

$$h^*(x) = \arg\max_{y \in \mathcal{Y}} P(y \mid x).$$

This is known as the Bayes optimal classifier. Note that $P(X, Y)$ is the true but unknown data distribution.

Proof. The expected 0/1 loss of a classifier $h(x)$ is

$$\begin{aligned}\mathbb{E}(L((X,Y),h)) &= \mathbb{E}_X\, \mathbb{E}_{Y|X}(I(Y \neq h(X))) \\ &= \mathbb{E}_X\, P(Y \neq h(X) \mid X).\end{aligned}$$

Hence we can set the value of $h(x)$ independently for each x by choosing it to minimize the expression under expectation. This leads to $h^*(x) = \arg\max_{y \in \mathcal{Y}} P(y \mid x)$.

Revisiting kNN, naive Bayes and logistic regression

- The Bayes optimal classifier requires knowing P, which is not provided.
- kNN, NB and LR all attempt to estimate $P(Y \mid X)$ from data.

Regression

- Given $D = \{(x_1, y_1), \ldots, (x_n, y_n)\} \subseteq \mathcal{X} \times \mathbf{R}$, find a function f that maps an input $x \in \mathcal{X}$ to $y \in \mathbf{R}$.
- We usually use the quadratic loss

$$L((x, y), h) = (h(x) - y)^2.$$

- ERM chooses a function with minimum mean squared error

$$\min_{h \in \mathcal{H}} \frac{1}{n} \sum_i (h(x_i) - y_i)^2.$$

Regression Function

The minimizer of the expected quadratic loss is the regression function

$$h^*(x) = \mathbb{E}(Y \mid x).$$

Proof. The expected quadratic loss of a function h is

$$\mathbb{E}((h(X) - Y)^2) = \mathbb{E}_X\big(h(X)^2 - 2h(X)\,\mathbb{E}(Y \mid X) + \mathbb{E}(Y^2 \mid X)\big).$$

Hence we can set the value of $h(x)$ independently for each x by choosing it to minimize the expression under expectation. This leads to $h^*(x) = \mathbb{E}(Y \mid x)$.

Revisiting kNN

- kNN's model approximates the regression function $h^*(x) = \mathbb{E}(Y \mid x)$.
- It uses the outputs for the k neighbors of x as an approximate sample from $P(Y \mid x)$, and uses the mean of this approximate sample to approximate $\mathbb{E}(Y \mid x)$.
- Under mild conditions, as $k \to \infty$ and $n/k \to \infty$, $h_n(x) \to h^*(x)$, for any distribution $P(X, Y)$.

Revisiting OLS (optional)

OLS as a special case of ERM

- OLS finds a hyperplane minimizing the sum of squared errors

$$\beta_n = \arg\min_{\beta \in \mathbf{R}^d} \sum_{i=1}^{n} (\mathbf{x}_i^\top \beta - y_i)^2.$$

- This is ERM where
  - The input set is $\mathcal{X} = \mathbf{R}^d$, and the output set is $\mathcal{Y} = \mathbf{R}$.
  - The hypothesis space are hyperplanes $\mathcal{H} = \{\mathbf{x}^\top \beta : \beta \in \mathbf{R}^d\}$.
  - Quadratic loss is used, as typically in regression.

Optimal hyperplane

Assume that $\mathbb{E}(XX^\top)$ is non-singular. The hyperplane $f^*(\mathbf{x}) = \beta^{*\top}\mathbf{x}$ with

$$\beta^* = \mathbb{E}(XX^\top)^{-1}\,\mathbb{E}(XY),$$

minimizes the expected quadratic loss among all hyperplanes of the form $f(\mathbf{x}) = \beta^\top \mathbf{x}$.

Proof. The expected quadratic loss of a hyperplane β is

$$\begin{aligned} R(\beta) &= \mathbb{E}((\beta^\top X - Y)^2) \\ &= \mathbb{E}(\beta^\top X X^\top \beta - 2\beta^\top XY + Y^2) \\ &= \beta^\top \mathbb{E}(XX^\top)\beta - 2\beta^\top \mathbb{E}(XY) + E(Y^2). \end{aligned}$$

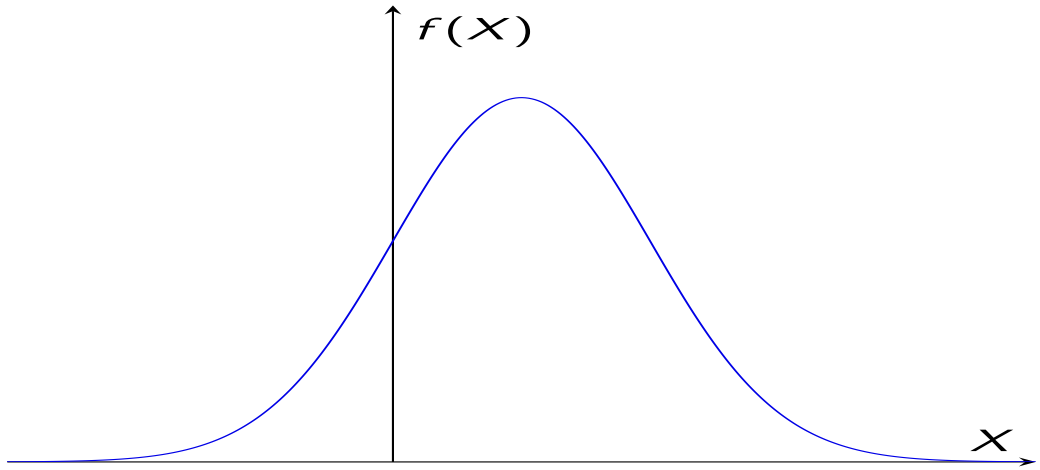
The gradient of R is

$$\nabla R(\beta) = 2\,\mathbb{E}(XX^\top)\beta - 2\,\mathbb{E}(XY).$$

Setting the gradient to 0 and solving for β, we have $\beta^* = \mathbb{E}(XX^\top)^{-1}\,\mathbb{E}(XY)$.

Density Estimation

E.g. learning a binomial distribution, or a Gaussian distribution.



We often use the log-loss

$$L(x, h) = -\ln p(x \mid h).$$

ERM is MLE in this case.

Analyzing the Performance of ERM (optional)

- The generalization error of ERM can be decomposed into three components
  - Estimation error arises from using a finite sample to find the best model.
  - Approximation error arises from using a hypothesis class not containing all functions.
  - Optimization error arises from using an optimization algorithm to approximately miminize the empirical risk.

Estimation error

- How does the empirically best hypothesis $h_n = \arg\min_{h \in \mathcal{H}} R_n(h)$ compare with the best in the hypothesis space? Specifically, how large is the estimation error $R(h_n) - \inf_{h \in \mathcal{H}} R(h)$?
- **Consistency**: Does $R(h_n)$ converge to $\inf_{h \in \mathcal{H}} R(h)$ as $n \to \infty$?

Estimation error

- How does the empirically best hypothesis $h_n = \arg\min_{h \in \mathcal{H}} R_n(h)$ compare with the best in the hypothesis space? Specifically, how large is the estimation error $R(h_n) - \inf_{h \in \mathcal{H}} R(h)$?
- **Consistency**: Does $R(h_n)$ converge to $\inf_{h \in \mathcal{H}} R(h)$ as $n \to \infty$?

If $|\mathcal{H}|$ is finite, ERM is likely to pick the function with minimal expected risk when n is *large*, because then $R_n(h)$ is close to $R(h)$ for all $h \in \mathcal{H}$.

If $|\mathcal{H}|$ is infinite, we can still show that ERM is likely to choose a near-optimal hypothesis if $\mathcal{H}$ has finite complexity (such as VC-dimension).

Approximation error

How good is the best hypothesis in $\mathcal{H}$? That is, how large is the approximation error $\inf_{h \in \mathcal{H}} R(h) - \inf_h R(h)$?

Approximation error

How good is the best hypothesis in $\mathcal{H}$? That is, how large is the approximation error $\inf_{h \in \mathcal{H}} R(h) - \inf_h R(h)$?

Trade-off between estimation error and approximation error:

- Larger hypothesis space implies smaller approximation error, but larger estimation error.
- Smaller hypothesis space implies larger approximation error, but smaller estimation error.

Optimization error

Is the optimization algorithm computing the empirically best hypothesis exactly?

Optimization error

Is the optimization algorithm computing the empirically best hypothesis exactly?

While ERM can be efficiently implemented in many cases, there are also computationally intractable cases, and efficient approximations are sought. The performance gap between the sub-optimal hypothesis and the empirically best hypothesis is the optimization error.

An Error Bound (optional)

If $|\mathcal{H}|$ is finite, and the loss function is bounded in $[0, 1]$, then with probability $1 - \delta$,

$$R(h_n) - \inf_{h \in \mathcal{H}} R(h) \leq 2\sqrt{\frac{\ln |\mathcal{H}| + \ln(2/\delta)}{2n}}$$

- The error decreases at a rate of $O(\frac{1}{\sqrt{n}})$.
- As $n \to \infty$, the RHS tends to 0, thus ERM is consistent in this case.

The case with $|\mathcal{H}| = \infty$

- Error bound depends on more sophisticated complexity measure (such as VC-dimension).
- The number of parameters does not measure the complexity of the model family.

What You Need to Know

- The statistical learning problem and ERM
- Classification, regression and density estimation as special cases of ERM.
- (Optional) The performance of ERM depends on the approximation error, estimation error and optimization error.

    error bounds can be derived using statistical theory