Supervised Learning via Empirical Risk Minimization

Sanjay Lall and Stephen Boyd

EE104
Stanford University
Predictors
Data fitting

—we think $y \in \mathbb{R}$ and $x \in \mathbb{R}^d$ are (approximately) related by

$$y \approx f(x)$$

—we don’t know the ‘true’ relationship between $x$ and $y$
Features

often $x$ is a vector of features:

- documents
  - $x$ is word count histogram for a document
- patient data
  - $x$ are patient attributes, test results, symptoms
- customers
  - $x$ is purchase history and other attributes of a customer
Where features come from

- we use $u$ to denote the raw input data, such as a vector, word or text, image, video, audio, ...

- $x = \phi(u)$ is the corresponding feature vector

- the function $\phi$ is called the embedding or feature function

- $\phi$ might be very simple or quite complicated

- similarly, the raw output data $v$ can be featurized as $y = \psi(v)$

- often we take $\phi(u)_1 = x_1 = 1$, the constant feature

- (much more on these ideas later)
we are given data $x^1, \ldots, x^n \in \mathbb{R}^d$ and $y^1, \ldots, y^n \in \mathbb{R}$

$(x^i, y^i)$ is the $i$th data pair or observation or example

we also (might) have prior knowledge about what $f$ might look like

- e.g., $f$ is smooth or continuous: $f(x) \approx f(\tilde{x})$ when $x$ is near $\tilde{x}$
- or we might know $y \geq 0$
we seek a *predictor* or *model* $g : \mathbb{R}^d \rightarrow \mathbb{R}$

for feature vector $x$, our prediction (of $y$) is $\hat{y} = g(x)$

predictor $g$ is chosen based on both data and prior knowledge

in terms of raw data, our predictor is

$$\hat{v} = \psi^{-1}(g(\phi(u)))$$

(with a slight variation when $\psi$ is not invertible)

$\hat{y}^i \approx y^i$ means our predictor does well on $i$th data pair

*but our real goal is to have* $\hat{y} \approx y$ *for* $(x, y)$ *pairs we have not seen*
Prediction methods

- fraud, psychic powers, telepathy, magic sticks, incantations, crystals, hunches, statistics, AI, machine learning, data science

- and many algorithms . . .

- example: nearest neighbor predictor
  - given \( x \), find its nearest neighbor \( x^i \) among given data
  - then predict \( \hat{y} = g(x) = y^i \)

A learning algorithm is a recipe for producing a predictor given data
Example: Nearest neighbor prediction

- left plot shows nearest neighbor prediction
- right plot shows fit with cubic polynomial
Linear predictors
Linear predictor

- predictors that are linear functions of $x$ are widely used
- a linear predictor has the form

$$g(x) = \theta^T x$$

for some vector $\theta \in \mathbb{R}^d$, called the predictor parameter vector

- also called a regression model
- $x_j$ is the $j$th feature, so the prediction is a linear combination of features

$$\hat{y} = g(x) = \theta_1 x_1 + \cdots + \theta_d x_d$$

- we get to choose the predictor parameter vector $\theta \in \mathbb{R}^d$
- sometimes we write $g_\theta(x)$ to emphasize the dependence on $\theta$
Interpreting a linear predictor

\[ \hat{y} = g(x) = \theta_1 x_1 + \cdots + \theta_d x_d \]

- \( \theta_3 \) is the amount that prediction \( \hat{y} = g(x) \) increases when \( x_3 \) increases by 1.
  - particularly interpretable when \( x_3 \) is Boolean (only takes values 0 or 1)
- \( \theta_7 = 0 \) means that the prediction does not depend on \( x_7 \)
- \( \theta \) small means predictor is insensitive to changes in \( x \):
  \[ |g(x) - g(\tilde{x})| = |\theta^T x - \theta^T \tilde{x}| = |\theta^T (x - \tilde{x})| \leq \|\theta\|_2 \|x - \tilde{x}\|_2 \]
Affine predictor

- suppose the first feature is constant, $x_1 = 1$
- the linear predictor $g$ is then an affine function of $x_{2:d}$, i.e., linear plus a constant
  \[ g(x) = \theta^T x = \theta_1 + \theta_2 x_2 + \cdots + \theta_d x_d \]
- $\theta_1$ is called the offset or constant term in the predictor
- $\theta_1$ is the prediction when all features (except the constant) are zero
Empirical risk minimization
**Loss function**

A *loss* or *risk* function $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ quantifies how well (more accurately, how badly) $\hat{y}$ approximates $y$.

- Smaller values of $\ell(\hat{y}, y)$ indicate that $\hat{y}$ is a good approximation of $y$.
- Typically (but not always) $\ell(y, y) = 0$ and $\ell(\hat{y}, y) \geq 0$ for all $\hat{y}, y$.

**Examples**

- **Quadratic loss**: $\ell(\hat{y}, y) = (\hat{y} - y)^2$
- **Absolute loss**: $\ell(\hat{y}, y) = |\hat{y} - y|$
**Empirical risk**

how well does the predictor \( g \) fit a data set \((x^i, y^i), i = 1, \ldots, n\), with loss \( \ell \)?

- the *empirical risk* is the average loss over the data points,

\[
\mathcal{L} = \frac{1}{n} \sum_{i=1}^{n} \ell(\hat{y}^i, y^i) = \frac{1}{n} \sum_{i=1}^{n} \ell(g(x^i), y^i)
\]

- if \( \mathcal{L} \) is small, the predictor predicts the given data well

- when the predictor is parametrized by \( \theta \), we write

\[
\mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(g_\theta(x^i), y^i)
\]

to show the dependence on the predictor parameter \( \theta \)
Mean square error

- for square loss \( \ell(\hat{y}, y) = (\hat{y} - y)^2 \), empirical risk is mean-square error (MSE)
  \[
  \mathcal{L} = \text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (g(x^i) - y^i)^2
  \]

- often we use root-mean-square error, \( \text{RMSE} = \sqrt{\text{MSE}} \), which has same units/scale as outcomes \( y^i \)
Mean absolute error

- For absolute value $\ell(\hat{y}, y) = |\hat{y} - y|$, empirical risk is mean-absolute error

$$\mathcal{L} = \frac{1}{n} \sum_{i=1}^{n} |g(x^i) - y^i|$$

- Has same units/scale as outcomes $y^i$

- Similar to, but not the same as, mean-square error
Empirical risk minimization

- choosing the parameter $\theta$ in a parametrized predictor $g_\theta(x)$ is called \textit{fitting the predictor} (to data)

- \textit{empirical risk minimization (ERM)} is a general method for fitting a parametrized predictor

- ERM: \textit{choose $\theta$ to minimize empirical risk $\mathcal{L}(\theta)$}

- thus, ERM chooses $\theta$ by attempting to match given data

- often there is no analytic solution to this minimization problem, so we use \textit{numerical optimization} to find $\theta$ that minimizes $\mathcal{L}(\theta)$ (more on this topic later)