Chapter 6
Mathematical Foundations of Supervised Learning
Overview

• Previously: we performed clustering
• Today: let us turn supervised
  – Fundamentals
  – Learning setup
  – Brief overview of learning algorithms
  – Our CrowdSignals case study
Supervised learning (1)

- What do we mean when we say "machines can learn"?
  - Task
  - Historical data
  - Improvement on task performance

- Mitchell: “A computer program is said to learn from experience E with respect to some class of tasks T and performance P, if its performance at tasks in T improves with E.”
Supervised learning (2)

• In supervised learning we want to learn
  – a functional relationship, alternatively called an unknown target function \( f : \mathcal{X} \rightarrow \mathcal{Y} \)

• Here we map an observation \( x \in \mathcal{X} \)
to the target \( y \in \mathcal{Y} \) (or \( g \))

• Here, \( \mathcal{X} \) represents the space of all possible inputs (and \( \mathcal{Y} \) all possible targets)

• Not likely to be deterministic
  – Measurement errors
  – Noisy target (e.g. because we do not have all observations)
Supervised learning (3)

• Assume unknown conditional target distribution $p(y|x)$
• Gives probability of $y$ given that $x$ is fixed
• If we calculate $f(x)$ we add noise from a noise distribution
  – Bernoulli or categorical distribution for discrete target
  – Normal distribution for continuous
• More difficult to learn the target function with this noise
• In addition we have the distribution of the observations $p(x)$
Supervised learning (4)

In almost all realistic cases the data will not be generated by a deterministic target function. That is because we often have measurement errors on the input side (e.g. when recording his food intake Bruce forgets unintentionally about the last piece of cream cake he had for breakfast) but also some noise on the output (e.g. we have not accounted for all factors that influence the glucose level). These two sources of noise would need to be dealt with independently, especially in cases of non-linear relations. However, as per usual in machine learning, we will restrict ourself and account only for noisy targets by introducing what we call the unknown conditional target distribution \( p(y|x) \). If you haven't seen the notation \( p(y|x) \) (read as the probability of \( y \) given \( x \)), do not be scared. It gives the probability distribution of \( y \) assuming that \( x \) is fixed. So, instead of having a deterministic function, we take an input value \( x \) calculate \( f(x) \) and then add some noise, generated from a noise distribution. When the target is discrete the noise distribution can be the Bernoulli or categorical distribution. In case of a numerical target as in the case of the glucose level it will be a continuous probability distribution such as the normal distribution. As you can imagine accounting for random noise makes it more tricky to (at least approximately) learn the target function, but that's what realistic datasets are like.

There is one piece missing to "generate" the observed data, which is the unknown input distribution \( p(x) \) - as you might expect it tells us how the observations are distributed. We will see later in this section why it is important to conceptually have \( p(x) \) as a building block of our learning setup even though we do not need to explicitly know it. Note, that we could now determine the joint distribution \( p(x, y) \),

---

![Supervised learning diagram](image-url)
Observed Data

- We separate our dataset into a training, validation, and test set
- We learn a function \( \hat{f}(x) \) that fits our observed data in the training set
- Should stratify training set in case of unbalanced dataset
- Evaluate generalizability of \( \hat{f}(x) \) upon test set
- Stop learning process based on validation set
- Small dataset: cross validation
Error measure (1)

- Assume we have a hypothesis for the target function $h$
- How far apart is $h$ from $f$? $E(f, h)$
  - risk
- We can compute it per point $e(f(x), h(x))$
  - loss
- From $e$ to $E$ (given $p(x)$):
  $$E(f, h) = \int_X e(f(x), h(x)) p(x) dx$$
• Could we computer this?
  – Nope
• We approximate it using the data we have
  \[ E(f, h) \approx \frac{1}{N} \sum_{j=1}^{N} e(y_j, h(x_j)) \]
• What definitions do we have for \( e \)?
  – Classification example: classification rate, F1, AUC, ….
  – Regression: mean squared error, mean absolute error, ….
Error measure (3)

- In sample error:
  \[ E_{\text{in}}(h) = \frac{1}{N} \sum_{(x,y) \in \mathcal{O}_{\text{Train}}} e(y, h(x)) \]

- Out of sample error:
  \[ E_{\text{out}}(h) = \int_{\mathcal{X} \setminus \mathcal{O}_{\text{Train}}} e(f(x), h(x)) p(x) dx \]
Hypotheses

• We select the hypothesis that minimizes the in-sample error:

\[ \hat{f} = \arg\min_{h \in \mathcal{H}} E_{in}(h) \]

• What is the set of hypotheses?
  – For linear regression:

\[ h_\theta(x_j) = \theta_0 + \sum_{k=1}^{p} \theta_k x_{jk} = \theta^T x_j \]
Model selection

• We select the hypothesis that has the lowest in-sample error on the validation set
• We should be careful about overfitting and not use too many features
Learning theory

• Are we able to learn the perfect model \((f)\)?
  – Even with infinite training examples the answer is often “no”
  – Why? Because we cannot guarantee that \(f \in \mathcal{H}\)

• Despite this, learning is possible if the in-sample error is a good estimator of the out-sample error \(|E_{out}(\hat{f}) - E_{in}(\hat{f})|\)

• Difference with higher \(N\) is most likely very small
PAC Learnable (1)

• We can make this a bit more formal:

Definition 6.2. A hypothesis set is said to be PAC learnable (probably approximately correct), if a learning algorithm exists that fulfills the following condition: For every $\varepsilon > 0$ and $\delta \in (0, 1)$ there is an $m \in \mathbb{N}$, so that for a random training sample with length larger than $m$, the following inequality holds with probability $1 - \delta$:

$$|E_{out}(\hat{f}) - E_{in}(\hat{f})| < \varepsilon$$

• Assume we have $M$ hypotheses, then:

$$E_{out}(\hat{f}) \leq E_{in}(\hat{f}) + \sqrt{\frac{1}{2N} \log \frac{2M}{\delta}}$$

• Hence, every finite set of hypotheses is PAC learnable
PAC Learnable (2)

- What if we have an infinite set of hypotheses?
- Let us focus on a binary classification problem and assume we have a function $h$ that is applied to all $N$ instances in our set:

**Definition 6.3.** Let the hypothesis set $\mathcal{H}$ be a set of functions $h : \mathcal{X} \rightarrow \{0, 1\}$. For a set of input vectors $X = \{x_1, x_2, \cdots, x_N\}$, we call $\mathcal{H}_X = \{(h(x_1), \cdots, h(x_N)) : h \in \mathcal{H}\}$ a restriction of $\mathcal{H}$ on $X$. 

\[ H \]
• How many hypotheses do we have?  
  – $2^N$
• A hypothesis set $\mathcal{H}$ shatters $X$ when it can represent every possible labeling (i.e. the $2^N$)
• We then define:

**Definition 6.4.** The Vapnik-Chervonenkis (VC) dimension $d_{VC}$ of a hypothesis set $\mathcal{H}$ is the maximum number of input vectors that can be shattered. The VC dimension is infinite if there are arbitrarily large sets of input vectors that can be shattered.
VC Dimension (2)

- We need to show this for an example, e.g. two real-valued attributes and the set of hypotheses involves a line
  - Can we represent all combinations for three points?

![Graphs showing hypotheses H1 and H2 separating three points.](image)

Definition 6.3. Let the hypothesis set $H$ be a set of functions $h : X \rightarrow \{0, 1\}$. For a set of input vectors $X = \{x_1, x_2, \ldots, x_N\}$, we call $H_X = \{(h(x_1), \ldots, h(x_N)) : h \in H\}$ a restriction of $H$ on $X$. An element of $H_X$ is built by taking a function $h \in H$ and applying it to all $N$ vectors in $X$. This yields a vector of length $N$ whose elements are 0's and 1's. Thus, $H_X$ contains a number of such length $N$ vectors. Since there are no more than $2N$ such vectors, the size of $H_X$ can maximally be $2^N$. However, the size of $H_X$ is often smaller than $2^N$, because various $h$ produce the same 0/1-vector. If the size of $H_X$ is $2^N$, we say $H$ shatters $X$, i.e. $H$ can realize every possible labeling of the $N$ input vectors. Based on that we define:

Definition 6.4. The Vapnik-Chervonenkis (VC) dimension $d_{VC}$ of a hypothesis set $H$ is the maximum number of input vectors that can be shattered. The VC dimension is infinite if there are arbitrarily large sets of input vectors that can be shattered. Note, that the VC dimension of a hypothesis set $d_{VC}$ does not impose that any set of input vectors that has size $d_{VC}$ can be shattered. Instead, the definition only requires the existence of at least one set of $d_{VC}$ input vectors that can be shattered. We admit that this might sound a bit complicated, so let's look at an example: Assume the input space $X$ is $\mathbb{R}^2$ and the hypothesis set $H$ encompasses all affine functions in $\mathbb{R}^2$, this is all straight lines. In Figure 6.5 you see a set of three points in $\mathbb{R}^2$. It is possible to find 8 different hypotheses (straight lines) that separate the three points in every of the $2^3$ ways. Can we shatter every set of three points by $H$? No, think of three points sitting on a straight line. Nevertheless, we have found a set of three points that can be shattered by $H$, so we know by definition that $d_{VC} = 3$. Can we choose 4 points in a way that they all can be shattered by $H$? As an exercise convince yourself that this is not possible. Therefore we know that $d_{VC} < 4$ and thus $d_{VC} = 3$. 

![Graphs showing hypotheses H1 and H2 separating four points.](image)
VC Dimension (3)

- Can we do it for 4?
  - Nope, VC-dimension is 3
- Now a nice result: all hypothesis sets with finite VC-dimension are PAC learnable
- In addition we can shown that:
  $$E_{out}(\hat{f}) \leq E_{in}(\hat{f}) + \sqrt{\frac{8}{N} \log \frac{4m_{\mathcal{H}}(2N)}{N}}$$
- With $m_{\mathcal{H}}(N)$ being the growth function
  - Measures the maximum number of elements for all possible restrictions $\mathcal{H}_X$
Implications (1)

- What are the implications?
  - With few training examples, it is easy to obtain a low in-sample error
  - As $N$ increases we will converge to a maximum
  - The out of sample error will decrease
Implications (2)

- Given a fixed $N$ we can study the influence of more complex hypotheses
Implications (3)

- Complexity versus training examples:

![Graph](image-url)

- In section 6.1.2 we briefly discussed the potential discrepancy between "best fitting function" and "good function". Given the discussion in section 6.1.5 how would you interpret this difference?

- In section 6.1.1 we mentioned that $f$ can be thought of as the generator of observed data. Discuss this statement and the perspective real world phenomena behind it.

- Why is $E_{\text{out}}(h) = E(f, h)$? In which cases does the equation break?

- Take the solution for linear regression $\hat{q} = (X^T X)^{-1} X^T Y$ and determine the dimensionality of the involved matrices and vectors.

- We stated that $-q E_{\text{in}}(h) = X^T (X q Y)$ for $N$. Now it is your turn to prove this equation. (Hint: it is tricky and needs some vector calculus. A good idea is to start with one training example and then extend the number of observations.)