Computational Learning Theory: An Analysis of a Conjunction Learner

Machine Learning
Spring 2018

The slides are mainly from Vivek Srikumar
This section

1. Analyze a simple algorithm for learning conjunctions
2. Define the PAC model of learning
3. Make formal connections to the principle of Occam’s razor
Learning Conjunctions

The true function \( f = x_2 \land x_3 \land x_4 \land x_5 \land x_{100} \)

Training data

- \( <(1,1,1,1,1,1,...,1,1), 1> \)
- \( <(1,1,1,0,0,0,...,0,0), 0> \)
- \( <(1,1,1,1,1,0,...,0,1,1), 1> \)
- \( <(1,0,1,1,1,0,...,0,1,1), 0> \)
- \( <(1,1,1,1,1,0,...,0,0,1), 1> \)
- \( <(1,0,1,0,0,0,...0,1,1), 0> \)
- \( <(1,1,1,1,1,1,...,0,1), 1> \)
- \( <(1,1,1,1,1,1,...,0,1), 1> \)
- \( <(0,1,0,1,0,0,...0,1,1), 0> \)
Learning Conjunctions

\[ f = x_2 \land x_3 \land x_4 \land x_5 \land x_{100} \]

Training data

- \( <(1,1,1,1,1,...,1,1), 1> \)
- \( <(1,1,1,0,0,...,0,0), 0> \)
- \( <(1,1,1,1,0,...0,1,1), 1> \)
- \( <(1,0,1,1,1,0,...0,1,1), 0> \)
- \( <(1,1,1,1,1,0,...0,0,1), 1> \)
- \( <(1,0,1,0,0,0,...0,1,1), 0> \)
- \( <(1,1,1,1,1,1,...,0,1), 1> \)
- \( <(0,1,0,1,0,0,...0,1,1), 0> \)

A simple learning algorithm (Elimination)

- Discard all negative examples
Learning Conjunctions

\[ f = x_2 \land x_3 \land x_4 \land x_5 \land x_{100} \]

Training data

- \((1,1,1,1,1,\ldots,1,1), 1\)
- \((1,1,1,0,0,0,\ldots,0,0), 0\)
- \((1,1,1,1,0,\ldots,0,1,1), 1\)
- \((1,0,1,1,1,0,\ldots,0,1,1), 0\)
- \((1,0,1,0,0,0,\ldots,0,0,1), 1\)
- \((1,0,1,0,0,0,\ldots,0,1,1), 0\)
- \((1,1,1,1,1,\ldots,0,1), 1\)
- \((1,1,1,1,1,\ldots,0,1), 1\)
- \((0,1,0,1,0,0,\ldots,0,1,1), 0\)

A simple learning algorithm (**Elimination**)

- Discard all negative examples
- Build a conjunction using the features that are common to all positive conjunctions

\[ h = x_1 \land x_2 \land x_3 \land x_4 \land x_5 \land x_{100} \]
Learning Conjunctions

\[ f = x_2 \land x_3 \land x_4 \land x_5 \land x_{100} \]

Training data

- \(<(1,1,1,1,1,1,...,1,1), 1>\>
- \(<(1,1,1,0,0,0,...,0,0), 0>\>
- \(<(1,1,1,1,0,...,0,1,1), 1>\>
- \(<(1,0,1,1,1,0,...,0,1,1), 0>\>
- \(<(1,1,1,1,1,0,...,0,0,1), 1>\>
- \(<(1,0,1,0,0,0,...,0,1,1), 0>\>
- \(<(1,1,1,1,1,1,...,0,1), 1>\>
- \(<(0,1,0,1,0,0,...,0,1,1), 0>\>

A simple learning algorithm (*Elimination*)

- Discard all negative examples
- Build a conjunction using the features that are common to all positive conjunctions

\[ h = x_1 \land x_2 \land x_3 \land x_4 \land x_5 \land x_{100} \]

Positive examples *eliminate* irrelevant features
Learning Conjunctions

\[ f = x_2 \land x_3 \land x_4 \land x_5 \land x_{100} \]

Training data

- \( <(1,1,1,1,1,1,...,1,1), 1> \)
- \( <(1,1,1,0,0,0,...,0,0), 0> \)
- \( <(1,1,1,1,1,0,...0,1,1), 1> \)
- \( <(1,0,1,1,1,0,...0,1,1), 0> \)
- \( <(1,1,1,1,1,0,...0,0,1), 1> \)
- \( <(1,0,1,0,0,0,...0,0,1), 0> \)
- \( <(1,0,1,1,1,1,...,0,1), 1> \)
- \( <(0,1,0,1,0,0,...0,1,1), 0> \)

A simple learning algorithm:

- Discard all negative examples
- Build a conjunction using the features that are common to all positive conjunctions

\[ h = x_1 \land x_2 \land x_3 \land x_4 \land x_5 \land x_{100} \]

Clearly this algorithm produces a conjunction that is consistent with the data, that is \( \text{err}_S(h) = 0 \), if the target function is a monotone conjunction

**Exercise:** Why?
Learning Conjunctions: Analysis

\[ f = x_2 \land x_3 \land x_4 \land x_5 \land x_{100} \]  \hspace{1cm} \[ h = x_1 \land x_2 \land x_3 \land x_4 \land x_5 \land x_{100} \]

**Claim 1:** \( h \) will only make mistakes on positive future examples

Why?
Learning Conjunctions: Analysis

\[ f = x_2 \land x_3 \land x_4 \land x_5 \land x_{100} \quad h = x_1 \land x_2 \land x_3 \land x_4 \land x_5 \land x_{100} \]

**Claim 1:** \( h \) will only make mistakes on positive future examples

*Why?*

A mistake will occur only if some feature \( z \) (in our example \( x_1 \)) is present in \( h \) but not in \( f \)

This mistake can cause a positive example to be predicted as negative by \( h \)

Specifically: \( x_1 = 0, x_2 = 1, x_3 = 1, x_4 = 1, x_5 = 1, x_{100} = 1 \)
Learning Conjunctions: Analysis

\[ f = x_2 \land x_3 \land x_4 \land x_5 \land x_{100} \quad h = x_1 \land x_2 \land x_3 \land x_4 \land x_5 \land x_{100} \]

Claim 1: h will only make mistakes on positive future examples

Why?

A mistake will occur only if some feature \( z \) (in our example \( x_1 \)) is present in \( h \) but not in \( f \)

This mistake can cause a positive example to be predicted as negative by \( h \)

Specifically: \( x_1 = 0, x_2 = 1, x_3 = 1, x_4 = 1, x_5 = 1, x_{100} = 1 \)

The reverse situation can never happen

For an example to be predicted as positive in the training set, every relevant feature must have been present
Learning Conjunctions: Analysis

\[ f = x_2 \land x_3 \land x_4 \land x_5 \land x_{100} \quad h = x_1 \land x_2 \land x_3 \land x_4 \land x_5 \land x_{100} \]

Claim 1: h will only make mistakes on positive future examples

Why?

A mistake will occur only if some feature \( z \) (in our example \( x_1 \)) is present in \( h \) but not in \( f \)

This mistake can cause a positive example to be predicted as negative by \( h \)

Specifically: \( x_1 = 0, x_2 = 1, x_3 = 1, x_4 = 1, x_5 = 1, x_{100} = 1 \)

The reverse situation can never happen

For an example to be predicted as positive in the training set, every relevant feature must have been present
Learning Conjunctions: Analysis

**Theorem**: Suppose we are learning a conjunctive concept with \( n \) dimensional Boolean features using \( m \) training examples. If

\[
m > \frac{n}{\epsilon} \left( \log(n) + \log \left( \frac{1}{\delta} \right) \right)
\]

then, with probability \( > 1 - \delta \), the error of the learned hypothesis \( \text{err}_D(h) \) will be less than \( \epsilon \).
Learning Conjunctions: Analysis

**Theorem:** Suppose we are learning a conjunctive concept with $n$ dimensional Boolean features using $m$ training examples. If

$$m > \frac{n}{\epsilon} \left( \log(n) + \log \left( \frac{1}{\delta} \right) \right)$$

then, with probability $> 1 - \delta$, the error of the learned hypothesis $\text{err}_D(h)$ will be less than $\epsilon$.

If we see these many training examples, then the algorithm will produce a conjunction that, with high probability, will make few errors.
Learning Conjunctions: Analysis

**Theorem:** Suppose we are learning a conjunctive concept with $n$ dimensional Boolean features using $m$ training examples. If

$$m > \frac{n}{\epsilon} \left( \log(n) + \log \left( \frac{1}{\delta} \right) \right)$$

then, with probability $> 1 - \delta$, the error of the learned hypothesis $err_D(h)$ will be less than $\epsilon$.

*Let’s prove this assertion*
Proof Intuition

\[ f = x_2 \land x_3 \land x_4 \land x_5 \land x_{100} \quad h = x_1 \land x_2 \land x_3 \land x_4 \land x_5 \land x_{100} \]

What kinds of examples would drive a hypothesis to make a mistake?
Proof Intuition

\[ f = x_2 \land x_3 \land x_4 \land x_5 \land x_{100} \quad h = x_1 \land x_2 \land x_3 \land x_4 \land x_5 \land x_{100} \]

What kinds of examples would drive a hypothesis to make a mistake?

Positive examples, where \( x_1 \) is absent
\( f \) would say true and \( h \) would say false
Proof Intuition

\[ f = x_2 \land x_3 \land x_4 \land x_5 \land x_{100} \quad h = x_1 \land x_2 \land x_3 \land x_4 \land x_5 \land x_{100} \]

What kinds of examples would drive a hypothesis to make a mistake?

Positive examples, where \( x_1 \) is absent
  \( f \) would say true and \( h \) would say false

None of these examples appeared during training
  Otherwise \( x_1 \) would have been eliminated
Proof Intuition

\[ f = x_2 \land x_3 \land x_4 \land x_5 \land x_{100} \quad h = x_1 \land x_2 \land x_3 \land x_4 \land x_5 \land x_{100} \]

What kinds of examples would drive a hypothesis to make a mistake?

Positive examples, where \( x_1 \) is absent
\( f \) would say true and \( h \) would say false

None of these examples appeared during training
Otherwise \( x_1 \) would have been eliminated

If they never appeared during training, maybe their appearance in the future would also be rare!
Proof Intuition

\[ f = x_2 \land x_3 \land x_4 \land x_5 \land x_{100} \]
\[ h = x_1 \land x_2 \land x_3 \land x_4 \land x_5 \land x_{100} \]

What kinds of examples would drive a hypothesis to make a mistake?

Positive examples, where \( x_1 \) is absent
\[ f \] would say \textit{true} and \( h \) would say \textit{false}

None of these examples appeared during training
Otherwise \( x_1 \) would have been eliminated

If they never appeared during training, maybe their appearance in the future would also be rare!

Let’s quantify our surprise at seeing such examples
Learning Conjunctions: Analysis

Let $p(z)$ be the probability that, in an example drawn from $D$, the feature $z$ is 0 but the example has a positive label.

- That is, after training is done, $p(z)$ is the probability that in a randomly drawn example, the feature $z$ causes a mistake.

- For any $z$ in the target function, $p(z) = 0$. 

Let $p(z)$ be the probability that, in an example drawn from $D$, the feature $z$ is 0 but the example has a positive label.

- That is, after training is done, $p(z)$ is the probability that in a randomly drawn example, the feature $z$ causes a mistake.
- For any $z$ in the target function, $p(z) = 0$.

Remember that there will only be mistakes on positive examples for this toy problem.
Learning Conjunctions: Analysis

Let $p(z)$ be the probability that, in an example drawn from $D$, the feature $z$ is 0 but the example has a positive label.

- That is, after training is done, $p(z)$ is the probability that in a randomly drawn example, the feature $z$ causes a mistake.
- For any $z$ in the target function, $p(z) = 0$

\[
f = x_2 \land x_3 \land x_4 \land x_5 \land x_{100}\]

\[
h = \text{boxed } x_1 \land x_2 \land x_3 \land x_4 \land x_5 \land x_{100}\]

Remember that there will only be mistakes on positive examples for this toy problem.
Learning Conjunctions: Analysis

Let $p(z)$ be the probability that, in an example drawn from $D$, the feature $z$ is 0 but the example has a positive label.

- That is, after training is done, $p(z)$ is the probability that in a randomly drawn example, the feature $z$ causes a mistake.
- For any $z$ in the target function, $p(z) = 0$.

$f = x_2 \land x_3 \land x_4 \land x_5 \land x_{100} \quad <(0,1,1,1,1,0,...0,1,1), 1>$

$h = \boxed{x_1} \land x_2 \land x_3 \land x_4 \land x_5 \land x_{100}$

Remember that there will only be mistakes on positive examples for this toy problem.
Learning Conjunctions: Analysis

Let $p(z)$ be the probability that, in an example drawn from $D$, the feature $z$ is 0 but the example has a positive label.

- That is, after training is done, $p(z)$ is the probability that in a randomly drawn example, the feature $z$ causes a mistake.
- For any $z$ in the target function, $p(z) = 0$.

$$f = x_2 \land x_3 \land x_4 \land x_5 \land x_{100}$$

$$h = x_1 \land x_2 \land x_3 \land x_4 \land x_5 \land x_{100}$$

Remember that there will only be mistakes on positive examples for this toy problem.

$p(x_1)$: Probability that this situation occurs.
Learning Conjunctions: Analysis

Let $p(z)$ be the probability that, in an example drawn from D, the feature $z$ is 0 but the example has a positive label.

- That is, after training is done, $p(z)$ is the probability that in a randomly drawn example, the feature $z$ causes a mistake.

- For any $z$ in the target function, $p(z) = 0$. 
Learning Conjunctions: Analysis

Let \( p(z) \) be the probability that, in an example drawn from \( D \), the feature \( z \) is 0 but the example has a positive label.

- That is, after training is done, \( p(z) \) is the probability that in a randomly drawn example, the feature \( z \) causes a mistake.

- For any \( z \) in the target function, \( p(z) = 0 \).

We know that \( \text{err}_D(h) \leq \sum_{z \in h} p(z) \).

Via direct application of the union bound.
Learning Conjunctions: Analysis

Let \( p(z) \) be the probability that, in an example drawn from \( D \), the feature \( z \) is 0 but the example has a positive label.

- That is, after training is done, \( p(z) \) is the probability that in a randomly drawn example, the feature \( z \) causes a mistake.

- For any \( z \) in the target function, \( p(z) = 0 \).

We know that \( \text{err}_D(h) \leq \sum_{z \in h} p(z) \).

Via direct application of the union bound.

**Union bound**
For a set of events, probability that at least one of them happens < the sum of the probabilities of the individual events.
Learning Conjunctions: Analysis

- Call a feature $z$ **bad** if $p(z) > \frac{\epsilon}{n}$
- Intuitively, a **bad feature** is one that has a significant probability of not appearing with a positive example
  - (And, if it appears in all positive training examples, it can cause errors)

If there are no bad features, then $\text{err}_D(h) < \epsilon$
Learning Conjunctions: Analysis

- Call a feature $z$ **bad** if $p(z) > \frac{\epsilon}{n}$
- Intuitively, a **bad feature** is one that has a significant probability of not appearing with a positive example
  - (And, if it appears in all positive training examples, it can cause errors)

If there are no bad features, then $\text{err}_D(h) < \epsilon$
- Why? Because $\text{err}_D(h) \leq \sum_{z \in h} p(z)$
Learning Conjunctions: Analysis

- Call a feature $z$ bad if $p(z) > \frac{\epsilon}{n}$
- Intuitively, a bad feature is one that has a significant probability of not appearing with a positive example
  - (And, if it appears in all positive training examples, it can cause errors)

If there are no bad features, then $\text{err}_D(h) < \epsilon$
- Why? Because $\text{err}_D(h) \leq \sum_{z \in h} p(z)$

Let us try to see when this will not happen
Learning Conjunctions: Analysis

• Call a feature z \textit{bad} if \( p(z) > \frac{\epsilon}{n} \)

• Intuitively, a \textit{bad feature} is one that has a significant probability of not appearing with a positive example
  – (And, if it appears in all positive training examples, it can cause errors)

What if there are bad features?
Learning Conjunctions: Analysis

• Call a feature \( z \) **bad** if \( p(z) > \frac{\epsilon}{n} \)

• Intuitively, a **bad feature** is one that has a significant probability of not appearing with a positive example
  – (And, if it appears in all positive training examples, it can cause errors)

What if there are bad features?

Let \( z \) be a bad feature
What is the probability that it will not be eliminated by one training example?

\( n = \) dimensionality
Learning Conjunctions: Analysis

\[ n = \text{dimensionality} \]

- Call a feature \( z \) **bad** if \( p(z) > \frac{\epsilon}{n} \)
- Intuitively, a **bad feature** is one that has a significant probability of not appearing with a positive example
  - (And, if it appears in all positive training examples, it can cause errors)

What if there are bad features?

Let \( z \) be a bad feature

What is the probability that it will not be eliminated by one training example?

\[
Pr(z \text{ survives one example}) = 1 - Pr(z \text{ is eliminated by one example}) = 1 - p(z) < 1 - \frac{\epsilon}{n}
\]
Learning Conjunctions: Analysis

- Call a feature \( z \) bad if \( p(z) > \frac{\epsilon}{n} \)
- Intuitively, a bad feature is one that has a significant probability of not appearing with a positive example
  - (And, if it appears in all positive training examples, it can cause errors)

What if there are bad features?

Let \( z \) be a bad feature

What is the probability that it will not be eliminated by one training example?

\[
Pr(z \text{ survives one example}) = 1 - Pr(z \text{ is eliminated by one example})
\]

\[
= 1 - p(z)
\]

\[
< 1 - \frac{\epsilon}{n}
\]
Learning Conjunctions: Analysis

What we know so far:

$$Pr(A \text{ bad feature is not eliminated by one example}) \leq 1 - \frac{\epsilon}{n}$$

$n = \text{dimensionality}$
Learning Conjunctions: Analysis

What we know so far:

\[ Pr(A \text{ bad feature is not eliminated by one example}) \leq 1 - \frac{\epsilon}{n} \]

But say we have \( m \) training examples. Then

\[ Pr(A \text{ bad feature survives } m \text{ examples}) \leq (1 - \frac{\epsilon}{n})^m \]
Learning Conjunctions: Analysis

What we know so far:

\[ Pr(A \text{ bad feature is not eliminated by one example}) \leq 1 - \frac{\epsilon}{n} \]

But say we have \( m \) training examples. Then

\[ Pr(A \text{ bad feature survives } m \text{ examples}) \leq (1 - \frac{\epsilon}{n})^m \]

There are at most \( n \) bad features. So

\[ Pr(\text{Any bad feature survives } m \text{ examples}) \leq n(1 - \frac{\epsilon}{n})^m \]
Learning Conjunctions: Analysis

\[ Pr(\text{Any bad feature survives } m \text{ examples}) \leq n \left(1 - \frac{\epsilon}{n}\right)^m \]

We want this probability to be small

Why? So that we can choose enough training examples so that the probability that any \( z \) survives all of them is less than some \( \delta \)
Learning Conjunctions: Analysis

\[ Pr(\text{Any bad feature survives m examples}) \leq n \left(1 - \frac{\epsilon}{n}\right)^m \]

We want this probability to be small.

Why? So that we can choose enough training examples so that the probability that any \( z \) survives all of them is less than some \( \delta \).

That is, we want \( n \left(1 - \frac{\epsilon}{n}\right)^m < \delta \).
Learning Conjunctions: Analysis

\[
\Pr(\text{Any bad feature survives } m \text{ examples}) \leq n \left(1 - \frac{\epsilon}{n}\right)^m
\]

We want this probability to be small.

Why? So that we can choose enough training examples so that the probability that any \( z \) survives all of them is less than some \( \delta \).

That is, we want

\[
n \left(1 - \frac{\epsilon}{n}\right)^m < \delta
\]

We know that \( 1 - x < e^{-x} \). So it is sufficient to require

\[
n e^{-\frac{me}{n}} < \delta
\]
Learning Conjunctions: Analysis

\[ Pr(\text{Any bad feature survives } m \text{ examples}) \leq n \left(1 - \frac{\epsilon}{n}\right)^m \]

We want this probability to be small

Why? So that we can choose enough training examples so that the probability that any \( z \) survives all of them is less than some \( \delta \)

That is, we want \( n \left(1 - \frac{\epsilon}{n}\right)^m < \delta \)

We know that \( 1 - x < e^{-x} \). So it is sufficient to require \( n e^{-\frac{m \epsilon}{n}} < \delta \)

Or equivalently,

\[ m > \frac{n}{\epsilon} \left( \log(n) + \log \left( \frac{1}{\delta} \right) \right) \]
Learning Conjunctions: Analysis

To guarantee a probability of failure (i.e., error $>\epsilon$) that is less than $\delta$, the number of examples we need is

$$m > \frac{n}{\epsilon} \left( \log(n) + \log \left( \frac{1}{\delta} \right) \right)$$

That is, if $m$ has this property, then

- With probability $1 - \delta$, there will be no bad features,
- Or equivalently, with probability $1 - \delta$, we will have $\text{err}_D(h) < \epsilon$
Learning Conjunctions: Analysis

To guarantee a probability of failure (i.e., error $>\epsilon$) that is less than $\delta$, the number of examples we need is

$$m > \frac{n}{\epsilon} \left( \log(n) + \log \left( \frac{1}{\delta} \right) \right)$$

That is, if $m$ has this property, then

- With probability $1 - \delta$, there will be no bad features,
- Or equivalently, with probability $1 - \delta$, we will have $err_D(h) < \epsilon$

What does this mean:

- If $\epsilon = 0.1$ and $\delta = 0.1$, then for $n = 100$, we need 6908 training examples
Learning Conjunctions: Analysis

To guarantee a probability of failure (i.e., error $>\varepsilon$) that is less than $\delta$, the number of examples we need is

$$m > \frac{n}{\varepsilon} \left( \log(n) + \log \left( \frac{1}{\delta} \right) \right)$$

That is, if $m$ has this property, then

- With probability $1 - \delta$, there will be no bad features,
- Or equivalently, with probability $1 - \delta$, we will have $\text{err}_D(h) < \varepsilon$

What does this mean:

- If $\varepsilon = 0.1$ and $\delta = 0.1$, then for $n = 100$, we need 6908 training examples
- If $\varepsilon = 0.1$ and $\delta = 0.1$, then for $n = 10$, we need only 461 examples
Learning Conjunctions: Analysis

To guarantee a probability of failure (i.e., error $>\epsilon$) that is less than $\delta$, the number of examples we need is

$$m > \frac{n}{\epsilon} \left( \log(n) + \log \left( \frac{1}{\delta} \right) \right)$$

That is, if $m$ has this property, then

- With probability $1 - \delta$, there will be no bad features,
- Or equivalently, with probability $1 - \delta$, we will have $\text{err}_D(h) < \epsilon$

What does this mean:

- If $\epsilon = 0.1$ and $\delta = 0.1$, then for $n = 100$, we need 6908 training examples
- If $\epsilon = 0.1$ and $\delta = 0.1$, then for $n = 10$, we need only 461 examples
- If $\epsilon = 0.1$ and $\delta = 0.01$, then for $n = 10$, we need only 691 examples

Poly in $n, 1/\delta, 1/\epsilon$
Learning Conjunctions: Analysis

To guarantee a probability of failure (i.e., error $> \epsilon$) that is less than $\delta$, the number of examples we need is

$$m > \frac{n}{\epsilon} \left( \log(n) + \log \left( \frac{1}{\delta} \right) \right)$$

That is, if $m$ has this property, then

- With probability $1 - \delta$, there will be no bad features,
- Or equivalently, with probability $1 - \delta$, we will have $\text{err}_D(h) < \epsilon$

What we have here is a PAC guarantee

Our algorithm is *Probably Approximately Correct*