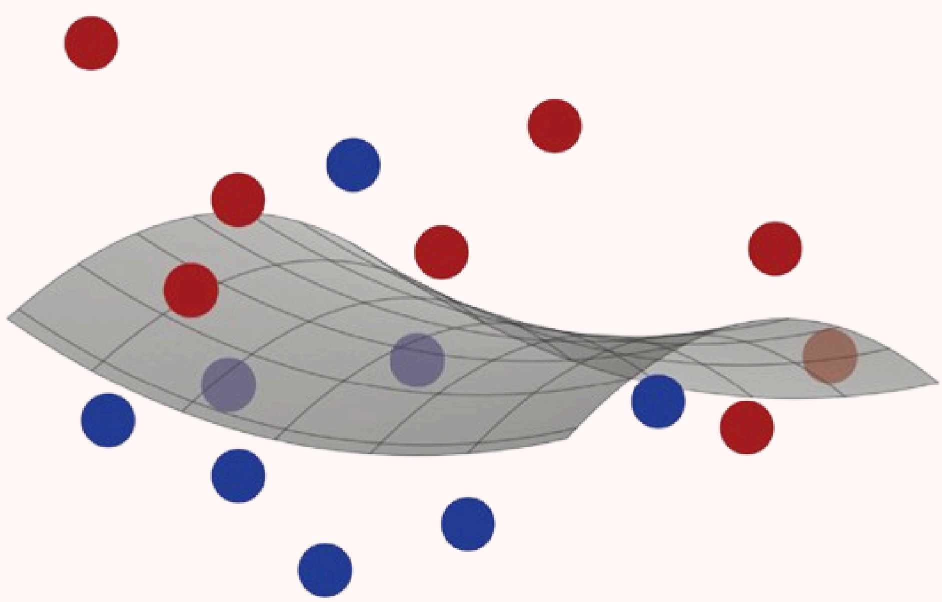
Foundations of Machine Learning

DAY - 9

Learning Guarantees for Finite Hypothesis Sets – Consistent Case



Learning Guarantees for Finite Hypothesis Sets – Consistent Case

Created By: **Birva Dave**
medium.com/@birva1809
github.com/Birva1809
linkedin.com/in/birva-dave
birvadave1809@gmail.com

When dealing with a finite hypothesis set H, if a learning algorithm always returns a consistent hypothesis (i.e., one that perfectly fits the training data), then it is possible to derive generalization guarantees based on the size of H and the number of training examples.

Key Result (Learning Bound for Consistent Hypotheses)

Let H be a finite set of functions mapping from input space X to output space Y. If a learning algorithm A, for any target concept $c \in H$, always returns a hypothesis $h_S \in H$ that is consistent with the training set S, then for any $\delta > 0$, the following holds:

- With probability at least $1-\delta$, the true error of h_S is bounded by:

$$R(h_S) \leq \frac{1}{m}\left(\log|H| + \log\frac{1}{\delta}\right)$$

This implies that the sample complexity, i.e., the number of training examples required to ensure the true error is at most with confidence, satisfies:

$$m \geq \frac{1}{\epsilon}\left(\log|H| + \log\frac{1}{\delta}\right)$$

Learning Guarantees for Finite Hypothesis Sets – Consistent Case

Created By: **Birva Dave**
medium.com/@birva1809
github.com/Birva1809
linkedin.com/in/birva-dave
birvadave1809@gmail.com

Proof Intuition

- Since the algorithm is consistent, $R_S(h_S) = 0$, and we are interested in bounding the probability that a bad hypothesis (with true error > ε) is consistent with the training data.

- The probability that a single bad hypothesis with error > ε is consistent on all m i.i.d. examples is at most $(1 - \varepsilon)^m$.

- Using the union bound over all such bad hypotheses in H, this total probability is at most $|H| * (1 - \varepsilon)^m$, which is $\leq |H| * e^{(-\varepsilon m)}$.

- Setting this less than or equal to δ and solving for m gives the desired sample complexity bound.

Implications

- A consistent algorithm over a finite hypothesis class is PAC-learnable.

- Larger hypothesis classes demand more data, but the dependency is only logarithmic in |H|.

- The logarithmic term log |H| can be seen as the number of bits required to represent the hypothesis set, emphasizing the trade-off between model complexity and sample size.

Learning Guarantees for Finite Hypothesis Sets – Consistent Case

Created By: **Birva Dave**
medium.com/@birva1809
github.com/Birva1809
linkedin.com/in/birva-dave
birvadave1809@gmail.com

Example: Conjunctions of Boolean Literals

- Consider learning conjunctions of up to nnn Boolean literals, where each literal is a variable x_i or its negation $\neg x_i$.
- The hypothesis space has size $|H| = 3^n$ since each variable can be included positively, negatively, or not at all.
- Using the general bound:

$$m \geq \frac{1}{\epsilon}\left(n \log 3 + \log \frac{1}{\delta}\right)$$

This shows that the class is PAC-learnable with sample size polynomial in n, 1/ϵ, log(1/δ). Computationally, the learning algorithm is efficient as it simply updates the valid literals based on positive examples.

Example: Universal Concept Class

- If X = 0, 1^n, then the universal concept class is the set of all subsets of X.
- This means $|H| = 4^n$, making the sample complexity:

$$m \geq \frac{1}{\epsilon}\left(2^n \log 2 + \log \frac{1}{\delta}\right)$$

This is exponential in n, meaning PAC-learning is not feasible. Even though consistency is possible, generalization is not guaranteed unless the hypothesis set is significantly smaller.

Learning Guarantees for Finite Hypothesis Sets – Consistent Case

Created By: **Birva Dave**
medium.com/@birva1809
github.com/Birva1809
linkedin.com/in/birva-dave
birvadave1809@gmail.com

Example: k-Term DNF Formulae

- A k-term DNF is a disjunction of k terms, each a conjunction of at most n literals.
- The size of the hypothesis space is $|H| = (3^n)^k$.
- Sample complexity:

$$m \geq \frac{1}{\epsilon}\left(kn\log 3 + \log\frac{1}{\delta}\right)$$

While this is polynomial in n, k, and 1/ε, efficient learning is unlikely unless RP = NP, due to a reduction from the graph 3-coloring problem. So, even though the sample complexity is reasonable, the computational complexity makes this class inefficient to learn.

Example: k-CNF Formulae

- A k-CNF is a conjunction of disjunctions, with each disjunction containing at most *k* literals.

- Using a clever variable mapping, learning k-CNF can be reduced to learning conjunctions of Boolean literals, which is PAC-learnable.

- Therefore, k-CNF formulae are PAC-learnable despite their expressive power.

- However, converting a learned k-CNF to an equivalent k-term DNF (even though such a conversion is theoretically possible) may not be computationally efficient unless RP = NP.

Learning Guarantees for Finite Hypothesis Sets – Consistent Case

Created By: **Birva Dave**
medium.com/@birva1809
github.com/Birva1809
linkedin.com/in/birva-dave
birvadave1809@gmail.com

Key Takeaways

- The PAC framework provides sample complexity guarantees for consistent learners over finite hypothesis sets.

- The generalization bound improves with more training examples and smaller hypothesis classes.

- There's a fundamental trade-off between the expressiveness of the hypothesis class and the feasibility of learning — both in terms of data and computation.

- A class may be PAC-learnable in theory but not efficiently PAC-learnable due to computational constraints.