Learning Theory for Classification

Patrick Martin

Department of Mathematics
Johns Hopkins University

Oral Exam - April 3, 2018

Outline

1. Introduction
2. Bounding the Estimation Error
   - Finite $\mathcal{C}$
   - Fingering
   - VC Dimension
3. Generative vs. Discriminative
   - Discriminative
   - Generative
   - Comparison of Error
4. Conclusion

What is Classification?

- Given some symptoms, what illness does a patient have?
- Given an English document, what dialect is it written in?
- Given an email, is it spam?
- Given some observables, what class produced it?

Giving some names

- Our observables will be feature vectors $x \in \mathbb{R}^d$
- Our classes will be numbers $i = 0, \dots, k-1$
- Thus our data is ordered pairs $(X, Y) \in \mathbb{R}^d \times \{0, \dots, k-1\}$
- A classifier is then a *function*

$$g : \mathbb{R}^d \to \{0, \dots, k-1\}$$

Data Generation - Part 1

- Our first paradigm of how the data is generated might be the following

Mixture model

Each class i represents a probability distribution over $\mathbb{R}^d$, and so data is generated by first choosing a class, and then generating a vector x based on that class's distribution $\mathcal{P}(x|i)$.

- Given a vector x, the probability that it belongs to a certain class is then given by Bayes' Rule:

$$\mathcal{P}(i|x) \propto \mathcal{P}(x|i)\,\mathcal{P}(i)$$

Data Generation - Part 2

- However, we really only care about the $\mathcal{P}(i|x)$, and so we can instead describe our data differently

Generative model

Data is generated by choosing vectors x, and then assigning a class i according to $\mathcal{P}(i|x)$.

Evaluation

- A classifier g makes an *error* for (X, Y) if $g(X) \neq Y$
- We can then talk about the *probability of error L*, in terms of the probability distribution ν:

$$L(g) := \mathcal{P}\left(g(X) \neq Y\right) = \nu\left(\{(X, Y) : \phi(X) \neq Y\}\right)$$

- The best classifier is the one that minimizes the probability of error

$$g^* = \arg\min_g L(g)$$

- This g^* is called the *Bayes classifier*, and $L^* := L(g^*)$ is the *Bayes error*

Ready to go?

- Unfortunately we can't really find g^* exactly, because
- The space of functions $g : \mathbb{R}^d \rightarrow \{0, \ldots, k-1\}$ is too large and unstructured to search
- We generally don't know $\mathcal{P}(x|i)$, and can't actually compute $L(g)$
- Instead, we will restrict our search to a subset of classifiers $\mathcal{C}$
- We will use training data to estimate the distributions
- For simplicity, we will also restrict ourselves to binary classification ($k = 1$)

Questions

- How well does the best classifier in our class do?
- How well can we approximate the best classifier?
- How much training data do we need to do this approximation?

Estimations and Approximations

- Introduce the *empirical* error probability for a classifier

$$\hat{L}_n(g) := \frac{1}{n}\sum_{i=1}^{n} \mathcal{I}_{g(X_i)\neq Y_i} = \nu_n(\{(X,Y) : \phi(X) \neq Y\})$$

- If we restrict our search to $\phi \in \mathcal{C}$, then we can choose ϕ_n^* to minimize $\hat{L}_n$
- How close to the Bayes error can we get?

$$L(\phi_n^*) - L^* = \underbrace{\left(L(\phi_n^*) - \inf_{\phi\in\mathcal{C}} L(\phi)\right)}_{\text{estimation error}} + \underbrace{\left(\inf_{\phi\in\mathcal{C}} L(\phi) - L^*\right)}_{\text{approximation error}}$$

- We can try to control the estimation error, but the approximation error belongs to $\mathcal{C}$

Finite $\mathcal{C}$

Lucky guess

If $|\mathcal{C}| < \infty$ and $\min_{\phi \in \mathcal{C}} L(\phi) = 0$, then for every n and $\varepsilon > 0$,

$$\mathcal{P}\left(L(\phi_n^*) > \varepsilon\right) \leq |\mathcal{C}|\, e^{-n\varepsilon}$$

and

$$\mathbb{E}\left(L(\phi_n^*)\right) \leq \frac{1 + \log|\mathcal{C}|}{n}$$

- However, this only holds if a “perfect” classifier is in our $\mathcal{C}$

Finite $\mathcal{C}$ - General case

- Turning our attention from L to ν, a classifier represents a subset of $\mathbb{R}^d \times \{0,1\}$

$$\phi \to A := \{(X,Y) : \phi(X) \neq Y\}$$

Unlucky guess

If a class of sets $\mathcal{A}$ has finite cardinality, then

$$\mathcal{P}\left(\sup_{A\in\mathcal{A}} |\nu_n(A) - \nu(A)| > \varepsilon\right) \leq 2\,|\mathcal{A}|\,e^{-2n\varepsilon^2}$$

- Notice that having a "perfect" classifier in $\mathcal{C}$ improves our bound!

Effectively finite $\mathcal{C}$

- While $\mathcal{C}$ will usually be infinite, it may be that there is a number k (the *fingering dimension*) and a function $\Psi : (\mathbb{R}^d)^k \to \mathcal{C}$ such that for any $x_1, \ldots, x_n$ the behavior of any $\phi \in \mathcal{C}$ can be replicated almost surely, with at most k mistakes, that is

$$\Psi(x_{i_1}, \ldots, x_{i_k})(x_j) = \phi(x_j) \quad \forall j \notin \{i_1, \ldots, i_k\}$$

- Fingering dimension of various types of classifiers
  - Linear classifiers: d
  - Hyperrectangular classifiers: $2d$
  - Spherical classifiers: $d + 1$
- In this case, for any training data we only have to look at finitely many classifiers, at most $\frac{n!}{(n-k)!}$ many.

Bounds

Bound on classifier selected by fingering

If $\mathcal{C}$ has fingering dimension k and $\hat{\phi}$ is found by fingering, then for $n \geq k$ and $\frac{2k}{n} \leq \varepsilon \leq 1$

$$\mathcal{P}\left(L(\hat{\phi}) - \inf_{\phi \in \mathcal{C}} L(\phi) > \varepsilon \right) \leq e^{2k\varepsilon}(n^k + 1)e^{-n\varepsilon^2/2}$$

and

$$\mathbb{E}\left(L(\hat{\phi}) - \inf_{\phi \in \mathcal{C}} L(\phi) \right) \leq \sqrt{\frac{(2k+1)\log n + (2k+2)}{n}}$$

Shattering Coefficient

- Another way of assigning a number to a class $\mathcal{C}$ is by looking at the shatter coefficient and the VC dimension
- Recall that a class of classifiers $\mathcal{C}$ induces a class of sets $\mathcal{A}$

Shatter Coefficient

The n-th *shatter coefficient* of a class of sets $\mathcal{A}$ is

$$S(\mathcal{C}, n) = s(\mathcal{A}, n) := \max_{(z_1,\ldots,z_n)\in(\mathbb{R}^d)^n} \left|\left\{\{z_1, \ldots, z_n\} \cap A_{0,1} : A \in \mathcal{A}\right\}\right|$$

Where $A_i := \{x \in \mathbb{R}^d : (x, i) \in A\}$.

- Immediately we see that $s(\mathcal{A}, n) \leq 2^n$

Bounds

Bound using Shatter Coefficient

$$\mathcal{P}\left(\sup_{\phi\in\mathcal{C}}\left|\hat{L}_n(\phi)-L(\phi)\right|>\varepsilon\right)\leq 8S(\mathcal{C},n)e^{-n\varepsilon^2/32}$$

Letting ϕ_n^* be a classifier minimizing $\hat{L}_n(\phi)$ over $\mathcal{C}$,

$$\mathcal{P}\left(L(\phi_n^*)-\inf_{\phi\in\mathcal{C}}(\phi)>\varepsilon\right)\leq 8S(\mathcal{C},n)e^{-2\varepsilon^2/128}$$

Also

$$\mathbb{E}\left(L(\phi_n^*)\right)-\inf_{\phi\in\mathcal{C}}L(\phi)\leq 16\sqrt{\frac{\log(8eS(\mathcal{C},n))}{2n}}$$

- Notice that this is only helpful if $S(\mathcal{C},n)\ll 2^n$

VC Dimension

- Let $k \geq 1$ be the largest integer such that

  $$s(\mathcal{A}, k) = 2^k$$

  This is the *VC dimension* of $\mathcal{A}$ (or $\mathcal{C}$), which will also be denoted $V_{\mathcal{A}}$ (or $V_{\mathcal{C}}$)

Bound on Shatter Coefficient

If $V_{\mathcal{C}} > 2$, then $S(\mathcal{C}, n) \leq n^{V_{\mathcal{C}}}$

- In other words, the shatter coefficient either grows exactly as 2^n, or is bounded by a polynomial.

Bounds

Bound using VC dimension

If $V_{\mathcal{C}} > 2$, then

$$\mathcal{P}\left(L(\phi_n^*) - \inf_{\phi \in \mathcal{C}} (\phi) > \varepsilon \right) \leq 8ne^{-2V_{\mathcal{C}}\varepsilon^2/128}$$

Also

$$\mathbb{E}\left(L(\phi_n^*)\right) - \inf_{\phi \in \mathcal{C}} L(\phi) \leq 16\sqrt{\frac{V_{\mathcal{C}} \log(n) + 4}{2n}}$$

Bounds

- In particular, this means that with high probability

$$L(\phi_n^*) - \inf_{\phi \in \mathcal{C}} L(\phi) \leq \mathcal{O}\left(\sqrt{\frac{V_{\mathcal{C}}}{n} \log n}\right)$$

- It turns out something a little stronger is true (Talagrand 1994):

$$L(\phi_n^*) - \inf_{\phi \in \mathcal{C}} L(\phi) \leq \mathcal{O}\left(\sqrt{\frac{V_{\mathcal{C}}}{n} \log \frac{n}{V_{\mathcal{C}}}}\right)$$

Classifier selection

- For a given algorithm of selecting a classifier from data, we also would like to know how much data we need to ensure a certain level of accuracy with certain confidence
- We say that $N(\varepsilon, \delta)$ is the *sample complexity* of an algorithm if it is the smallest integer such that if $n \geq N(\varepsilon, \delta)$, then if g_n is the selected classifier,

$$\sup_{(X,Y)} \mathcal{P}\left(L(g_n) - \inf_{\phi \in \mathcal{C}} L(\phi) > \varepsilon\right) \leq \delta$$

whenever $n \geq N(\varepsilon, \delta)$

Sample complexity of Empirical Risk Minimization

$$N(\varepsilon, \delta) \leq \max\left(\frac{512 V_{\mathcal{C}}}{\varepsilon^2} \log \frac{256 V_{\mathcal{C}}}{\varepsilon^2}, \frac{256}{\varepsilon^2} \log \frac{8}{\delta}\right)$$

Discriminative classifiers

- So far we have been looking at classifiers that are trying to approximate the Bayes classifier $\mathcal{P}\left(Y|X\right)$, as

$$\hat{L}_n(\phi) := \frac{1}{n}\sum_{i=1}^{n}\mathcal{I}_{\phi(X_i)\neq Y_i}$$

- These are called *discriminative classifiers*

Logistic Regression

- For example, let $\mathcal{C}$ be the class of linear classifiers
- Then one discriminative model is logistic regression

Logistic Regression

$$\mathcal{P}\left(Y = 1|X; \beta, \theta\right) = \frac{1}{1 + exp(\langle \beta, X \rangle + \theta)}$$

We can write a discriminant for this classifier:

$$D(X) = \langle \beta, X \rangle + \theta = \sum_{i=1}^{d} \beta_i X_i + \theta$$

where 1 is chosen if $D(X) > 0$, and 0 otherwise.

Convergence for logistic regression

- Let ϕ_n minimize $\hat{L}_n(\phi)$
- We have all this machinery for bounding convergence of linear models, so we can say that with high probability,

$$L(\phi_n) - \inf_{\phi \in \mathcal{C}} \leq \mathcal{O}\left(\sqrt{\frac{V_{\mathcal{C}}}{n} \log \frac{n}{V_{\mathcal{C}}}}\right)$$

VC dimension of linear classifiers

If $\mathcal{C}$ is the class of linear classifiers in $\mathbb{R}^d$, then $V_{\mathcal{C}} = d + 1$.

- Thus the rate of convergence is slightly worse than $\sqrt{\frac{1}{n}}$

Generative classifiers

- We could also select a classifier that optimizes some other function, for example one that tries to approximate the joint distribution $\mathcal{P}(X, Y) = \mathcal{P}(X|Y) \cdot \mathcal{P}(Y)$
- Thinking back to our first idea of how the data was generated, this is precisely trying to model that, and so these models are called *generative classifiers*, and compute $\mathcal{P}(Y|X)$ from $\mathcal{P}(X, Y)$, typically in a Bayesian way

Naive Bayes

- If we make a strong (naive) independence assumption amongst the features, we can then say that

Naive Bayes

$$\mathcal{P}(Y=1|X) \propto \mathcal{P}(Y=1)\prod_{i=1}^{d}\mathcal{P}(X_i|Y=1)$$

Equivalently, naive Bayes decides to label 1 if the following is greater than 1:

$$\frac{\mathcal{P}(Y=1|X)}{\mathcal{P}(Y=0|X)} = \frac{\mathcal{P}(Y=1)}{\mathcal{P}(Y=0)}\prod_{i=1}^{d}\frac{\mathcal{P}(X_i|Y=1)}{\mathcal{P}(X_i|Y=0)}$$

A linear classifier?

- Notice that also equivalently, the discriminant for naive Bayes is:

$$D(x) = \log \frac{\mathcal{P}(Y=1|X)}{\mathcal{P}(Y=0|X)} = \underbrace{\log \frac{\mathcal{P}(Y=1)}{\mathcal{P}(Y=0)}}_{\theta?} + \underbrace{\sum_{i=1}^{d} \log \frac{\mathcal{P}(X_i|Y=1)}{\mathcal{P}(X_i|Y=0)}}_{\langle \beta, X \rangle ?}$$

- This looks a lot like the definition of a linear classifier, and indeed naive Bayes and logistic regression are a *generative-discriminative pair*, in that abstractly they differ only in what they attempt to approximate

Error for Naive Bayes

- Let ϕ_n be a logistic regression on n datapoints, and ψ_n naive Bayes on n datapoints
- Simply due to the difference between generative and discriminative classifiers,

$$\hat{L}(\phi_n) \leq \hat{L}(\psi_n)$$

- Additionally, letting ϕ^* and ψ^* be the appropriate classifiers trained on the entire population, then

$$L(\phi^*) \leq L(\psi^*)$$

- However, we might want to know the convergence rate for naive Bayes as well

Convergence for Naive Bayes

Convergence for Naive Bayes (Ng and Jordan, 2002

Assume $\rho_0 \leq \mathcal{P}(Y = 1) \leq 1 - \rho_0$, and $Var(X_i) \geq \rho_0$ for $i = 1, \ldots, d$. Then, with high probability,

$$L(\psi_n) \leq L(\psi^*) + G\left(\mathcal{O}\left(\sqrt{\frac{1}{n}\log d}\right)\right)$$

where

$$G(\tau) = \mathcal{P}\left(\{(X,1) : D_{\psi^*}(X) \in [0, d\tau]\} \cup \{(X,0) : D_{\psi^*}(X) \in [-d\tau, 0]\}\right)$$

in other words, the probability of the population classifier coming within $d\tau$ of making an error

- Thus if $G(\tau) \leq \mathcal{O}(\tau)$, we have a convergence rate of $\sqrt{\frac{1}{}}$

Tradeoff

- Asymptotically, logistic regression performs better than naive Bayes as a classifier
- However, for small n, naive Bayes might actually do better, due to possibly faster convergence

Conclusions

- We would like to be able to bound the error of the classifier we choose from a class $\mathcal{C}$, and we can do so...
  - ...when $|\mathcal{C}|$ is finite
  - ...when $\mathcal{C}$ has finite fingering dimension
  - ...when $\mathcal{C}$ has finite VC dimension

Counter intuition

- However, discriminative classifiers are not the only type
- It turns out that while generative classifiers are not explicitly trained to minimize errors, for smallish n they can out-perform discriminative classifiers