Course review

Tuesday, November 26, 2019 11:29

Agenda: - Supervised learning
- Unsupervised learning
} all algos we covered pros/cons

Supervised Learning

Get examples $X_1, Y_1, \dots, X_n, Y_n$

Want to find $\hat{f}(X) \approx Y$ diff ways to express ≈

Where X, Y is a new unseen example
e.g. like the next unseen training examples X_{n+1}, Y_{n+1}

- Regression: $Y_i \in \mathbb{R}$
  - ≈ was usually measured using
    $\ell(\hat{f}(X), Y) = (\hat{f}(X) - Y)^2$
- Classification: $Y_i \in \{1, \dots, m\}$ discrete classes
  - ≈ usually measured using
    $\ell(\hat{f}(X), Y) = \mathbb{I}[\hat{f}(X) \neq Y]$

Classification often solved by
Probabilistic modelling
Want to find $\hat{p}(j|X) \approx \mathbb{P}(Y = j \mid X)$

Use for classification: $\hat{f}(X) = \underset{j=1,\dots,m}{\operatorname{argmax}} \hat{p}(j|X)$

≈ usually measured as
$\ell(\hat{p}(\cdot|X), Y) = -\log \hat{p}(Y|X)$
= neg log of prob assigned to the true label Y

Algs	Pros	Cons	When to use
kNN	simple, flexible, lazy	curse of dim needs $n \gg 2^p$ slow to compute	Very seldom
OLS & logistic	Simple, interpretable, linear fits a lot of things well, logistic reg gives a prob prediction	linear sometimes a bad fit (nothing is exactly linear) need to engineer features to go beyond linearity	First thing to try (given $n > p$)
Regularized OLS/ logistic (Ridge/Lasso)	- Can handle $p > n$ - tradeoff bias for variance which can give lower overall error - does variable selection	- need to choose hyperparam λ (or - can use CV)	<-//- for smaller n Want to discover

	(Lasso) Which improves interpretability	— — same	"important" features
Kernel Regression	Same as kNN + get smooth fns	Same as kNN	p=1 maybe p=2 never p≥3
Naive Bayes	Simple, interpretable handle large p n≈p	Naive assumptions bad fit Need to pick marginal learners	Good to try for hi-dim binary data E.g. Bag of words
CART	interpretable, flexible, auto var selection	Not always good fit Not best performance	Want an interpretable decision rule
RF	Flexible, very few tuning params, often just performs well out of the box	Hard to interpret Can be outdone by boosted trees well-engineered models	Definitely try if if linear is not enough
Boosting	Flexible, performs well in many examples, few tuning params	-Not many -A bit more complicated -Slightly more burdensome computationally - Can be outdone by well-engineered NNs	— // — & want improvement over RF
SVM	linear but easily kernelizable margin - robustness often better than logistic	- Unlike logistic, don't get prob output (can fix w/ platt scaling) - need to pick kernel - need to engineer features	Try for improvement over logistic
Feed-fwd NNs	- Very flexible - State of the art in cognition tasks (e.g. vision) - Don't need to engineer features	- Need to engineer architecture - Need to engineer learning scheme: which SGD algo, learning rate, weight decay - High overhead to just "try it out"	- Complex hi-dim inputs (e.g. pixels in an image) - All else fails or want improvement - Willing to code more to fiddle more - Know of domain success e.g. CNNs for vision