Cross validation:

- Hold out approach: for testing or approximating the true error of a classification.
- Let's assume classification; so hypothesis h is going to output $\{0,1\}$ or $\{-1,1\}$ values.

"Hold out": 1) Leave some part of training set out during training time.

2) Test classifier on this held-out set. Fraction of mistakes the estimate of the true error.

Review: Markov Inequality: let X be R.V. that takes on only positive values.

$$P[X \geq K \cdot E[X]] \leq \frac{1}{K}$$

Chebyshev's Inequality:

$$Var[X] = E(X - E[X])^2 = \left(\text{how much deviation from the mean}\right)^2$$

$$\sqrt{Var[X]} = STD(X) = \sigma$$

$$Pr[|X-\mu| > t\sigma] \leq \frac{1}{t^2}$$

☆ Chernoff Bound: $P(X \geq a) \leq \min_{s>0} e^{-sa} M_X(s)$.

Let $X_1, X_2, \dots X_n$ be i.i.d random variables.

Let $E[X_i] = p$

Let $S = \sum_{i=1}^{n} x_i$

Let $\mu = E[S] = n \cdot p$ (linearity of expectation) $E[X_1 + \dots X_n] = p \cdot n$

1) $Pr[S > \mu + \delta n] \leq e^{-2n\delta^2}$

2) $Pr[S < \mu - \delta n] \leq e^{-2n\delta^2}$

$\Rightarrow Pr[|S-\mu| > \delta n] \leq 2 \cdot e^{-2n\delta^2}$

Probability of deviating by more than δn is exponentially small in n, and depending on my choice of δ, will get different bounds.

* Apply the Chernoff Bound to the case of estimating the true error of a classifier.

Hold-out set S, with $|S| = n$,

Fix h (generated using some training set, which is independent from S, the hold-out set).

$\uparrow$ classifier

Recall $\mathcal{D}$ (distribution from which we generate training points),

S is a sample drawn from $\mathcal{D}$, independent of the training set.

$$z = \Pr_{\underbrace{x \sim \mathcal{D}}_{x \text{ drawn from } \mathcal{D}}} [h(x) \neq \underbrace{c(x)}] = \text{true error of classifier.}$$

↳ unknown function trying to learn.

How to estimate z?

Let X_i be R.V. that equals:

1 if h is incorrect on the i^{th} element of S, and

0 if h is correct on the i^{th} element of S.

RV: $X_1, \dots, X_n$. $X_i = \begin{cases} 1 & \text{if } h \text{ is incorrect on } i^{th} \text{ example} \\ 0 & \text{otherwise.} \end{cases}$

$S = \sum_{i=1}^{n} X_i$ $\qquad E[S] = n \cdot p$.

↖ p is the true error of h.

$$p = E[X_i] = E[X_1] = \dots = E[X_n]$$

$$\Pr[|S - n \cdot p| > \delta n] \leq 2e^{-2n\delta^2}$$

(Recall: p is the true error of classifier h).

$\delta = 0.1$

$$P\left[\,|S - n \cdot p| > 0.1\, n\,\right] \leq 2e^{-2n\left(\frac{1}{100}\right)}$$

How large to choose n before the quantity becomes small?

$$2e^{-2n\left(\frac{1}{100}\right)} = \alpha$$

$$e^{-2n/100} < \alpha/2$$

$$-\frac{2n}{100} < \ln\left(\frac{2}{\alpha}\right) \Rightarrow n > 50 \ln\left(\frac{2}{\alpha}\right)$$

If we want probability of failure to be $< \alpha$, and we want confidence that error estimate is $0.1\, n$, then we need n to be $50 \log\left(\frac{2}{\alpha}\right)$.

If $|S - n \cdot p| \leq 0.1 \cdot n \Rightarrow$ error rate on the hold-out is within 0.1 of the true error rate, with α confidence. (probability of at least $1 - \alpha$).

$\Rightarrow$ Chernoff Bound says: The # of samples in hold-out set, S, has to be $> 50 \ln\left(\frac{2}{\alpha}\right)$ if you want probability of [error rate on S larger than 0.1 of true error rate] to be at most α (i.e. with confidence $1 - \alpha$).

Hold-out set is somewhat expensive,

- Data is expensive.
- If we want to try out multiple methods for generating classifiers, we quickly lose confidence in our estimates. bc we have to add up the probabilities of failure. (if I generate another classifier and use the hold-out set again, the $P[\text{failure}] = 2\alpha$, ... and so on).

How can we reuse the training sets to build lots of different classifier and still understand their true error? Cross validation!

Not too much theory to explain why CV works well.

Cross-Validation : Algorithm.

1) train using Folds 2 ... Fold K. Hold out fold 1.
2) test on fold 1. Get the error.

Fold 1	Fold 2				Fold K

testing set → estimate the true error of classifer.

training set.

3) Hold out Fold 2, and train using folds 1, 3. ... n.
4) test on fold 2. Get the error.
5) Repeat, holding out fold k, and testing on fold k.
6) Take the average of the errors.

Let's go back to decision trees: we have a training set S.

· Should I build a decision tree of depth 10 or depth 15?

★ Decide using Cross - validation!

if just training error, then pick 15. But may overfit. Generalization error might be higher in depth 15.

Depth 10

Fold 1	Fold 2				Fold K

Depth 15

Fold 1	Fold 2				Fold K

whichever error is smaller is the one I would use.

For k, pick usually btwn 5 and 10 (no hard guideline).

In practice it works well. But difficult to say anything/analyze as this is not independent (the epochs).