Post- Pruning Cont.

AW

Lecture Overview

Reality of Bounding Errors: 1-or-2 Node DT classifiers

We are given a sample of size N.

The problem setting: What we know is that

- Single node tree classifier has associated Bernoulli distribution of correct classification: e if incorrect and $1 - e$ otherwise where e is unknown error
- A sample (estimated) error of this classifier is Y/N where Y is the number of incorrect classifications, and N is sample size. So we can assume that estimated error approaches e as the sample size increases (i.e. $N \to \infty$)
- In a two level classifier each leaf is a single node classifier, so everything above applies to leafs

What we need is:

1. for a single node classifier bound e with a given confidence δ ;
2. For a two level binary tree classifier bound combined leaf error (some combinations of e_1 and e_2 that we need to define) with a given confidence δ.

Node Model

- With a single node is associated Bernoulli distribution $\mathscr{B}(e, 1-e)$ of incorrect classification, i.e. e if incorrect and $1-e$ otherwise, e is unknown
- How are results of classification procedure distributed? Let $S = \{x_1, \ldots, x_N\}$ be data sample associated with the node. Since the nodes are classifying by majority, so for each x_i we have a true label l and assigned label y. Let $X_i = \begin{cases} 1 \text{ if } l \neq y \\ 0 \text{ otherwise} \end{cases}$. Obviously, X_i is a random variable distributed with $\mathscr{B}(e, 1-e)$,
- Sine $X_i \sim \mathscr{B}(e, 1-e)$ we have expectation $E(X_i) = e$ and variance $\sigma^2(X_i) = e(1-e)$,
- Let $Y = \sum_{i=1}^{N} X_i$. It is a random variable that counts the number of errors in sample S,
- So what is the probability that $Y = t$ for some $0 \leq t \leq N$?

Node Model

- With a single node is associated Bernoulli distribution $\mathscr{B}(e, 1-e)$ of incorrect classification, i.e. e if incorrect and $1-e$ otherwise, e is unknown
- How are results of classification procedure distributed? Let $S = \{x_1, \ldots, x_N\}$ be data sample associated with the node. Since the nodes are classifying by majority, so for each x_i we have a true label l and assigned label y. Let $X_i = \begin{cases} 1 \text{ if } l \neq y \\ 0 \text{ otherwise} \end{cases}$. Obviously, X_i is a random variable distributed with $\mathscr{B}(e, 1-e)$,
- Sine $X_i \sim \mathscr{B}(e, 1-e)$ we have expectation $E(X_i) = e$ and variance $\sigma^2(X_i) = e(1-e)$,
- Let $Y = \sum_{i=1}^{N} X_i$. It is a random variable that counts the number of errors in sample S,
- So what is the probability that $Y = t$ for some $0 \leq t \leq N$? It is $\binom{N}{t} e^t (1-e)^{N-t}$, so Y is binomially distributed. This is denoted $Y \sim Bin(e, N)$.

Expectation and Variance of Errors

- We know: $Y = \sum_{i=1}^{N} X_i$ where X_i, X_j are pairwise independent random variables (sampled) from $X_i \sim \mathscr{B}(e, 1-e)$ and $Y \sim Bin(e, N)$
- Expectation of errors: $E(Y) = \sum_{i=1}^{N} e = Ne$
- Variance of number of errors: $\sigma^2(Y) = Ne(1-e)$ (since X_i, X_j are pairwise independent random variables $Cov(X_i, X_j) = 0$)
- Frequency of errors $F = Y/N \sim Y$ since N is fixed,so $F \sim Bin(e, N)$.
  - Expectation and Variance of F:
    $E(F) = E(Y/N) = E(Y)/N = Ne/N = e$ and $\sigma^2(F) = E\left((F - E(F))^2\right) = E\left(\frac{(Y-E(Y))^2}{N^2}\right) = \frac{\sigma^2(Y)}{N^2} = \frac{Ne(1-e)}{N^2} = \frac{e(1-e)}{N}$

Road Map for Bounding e

- By Law of Large Numbers sample frequency F must converge to its expectation as $N \to \infty$. So given our sample S, let's try to bound the difference between F and e with given probability δ. In other words, let's find $b(F, \delta)$ such that $\Pr(|F - e| \leq b(F, \delta)) = \delta$.
- There are actually 2 events in $|F - e| \leq b(F, \delta)$: one is $F - e \leq b(F, \delta)$ and another is $e - F \leq b(F, \delta)$. The first event is of no use to us since it doesn't give upper bound on e (only lower bound). The second event is the one we are after: it gives upper bound $e \leq F + b(F, \delta)$.
- We can use symmetry of binomial distribution to find $b(F, \delta)$ such that $\Pr(e - F \leq b(F, \delta/2)) = \delta/2$. Then knowing F from sample and $b(F, \delta)$ we can bound e by a function of F,δ
- It is hard to manipulate with binomial distribution, so let's approximate/bound it using normal (thanks to De Moivre-Laplace theorem) and then carry out the first idea with normal distribution.

  Technicality: If we are going to work with normal distribution then it is convenient to express bounds in terms of tails as it is how normal is tabulated and returned by $\mathbf{R}$, so instead of δ we'll use $\alpha = 1 - \delta$

Binomial is Pretty Close to Normal

Theorem (De Moivre-Laplace Limit Theorem)

As $n$grows large, for $k \sim np$ we can approximate

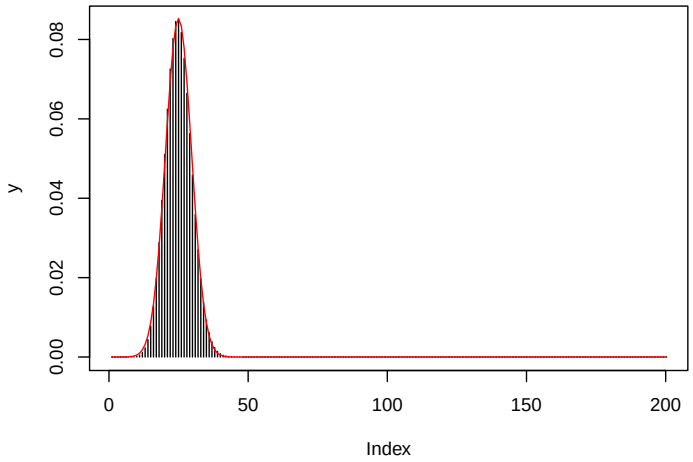
$$\binom{n}{k} p^k q^{n-k} \approx \frac{1}{\sqrt{2\pi npq}} e^{-\frac{(k-np)^2}{2npq}}, \text{ where } p+q=1,\ p,q>0$$

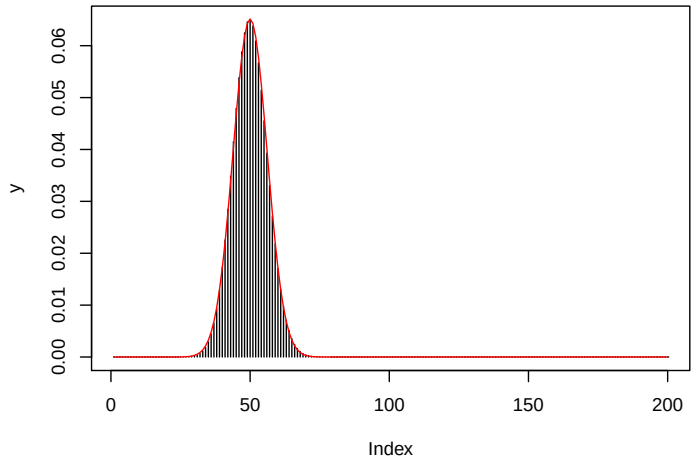
in the sense that the ratio of the left-hand side to the right-hand side converges to 1 *as* $n \to \infty$.

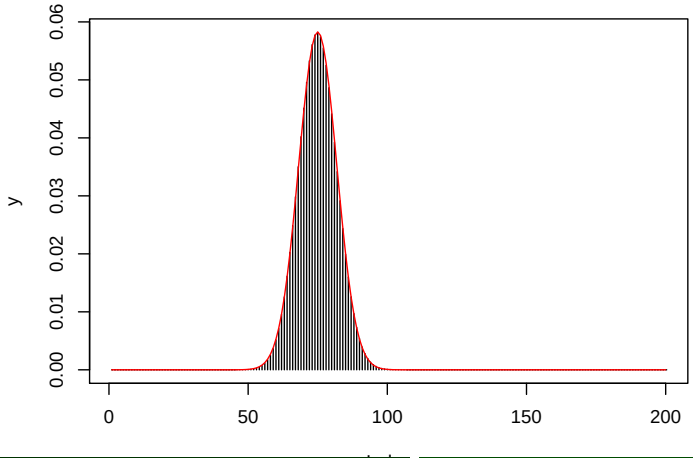
How good is this approximation?

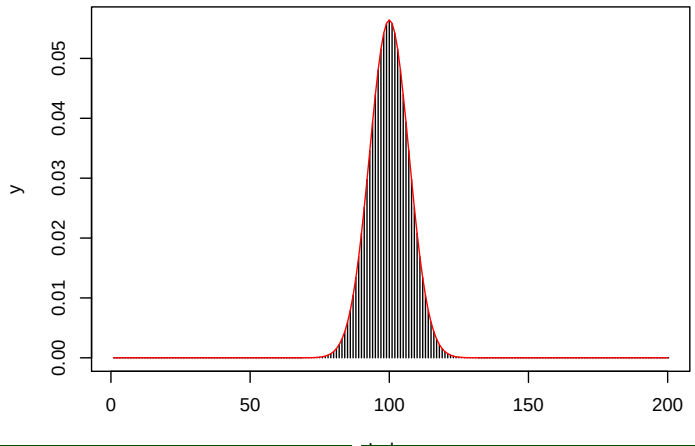
Binomial is Pretty Close to Normal

As long as $np > 5$ and $n(1-p) > 5$ it is pretty good, which means that for all but extreme values (e.g. $p < 0.01$) 500 samples is enough.









Lecture Overview

One-Node Tree Error Estimation

- Approximate $F = \frac{Y}{N} \sim Bin(e, N)$ with $F \sim \mathcal{N}(\mu, \sigma^2)$ where $\mu = E(F) = e$ and $\sigma^2 = Var(\frac{Y}{N}) = \frac{e(1-e)}{N}$
- Standardize $F \sim \mathcal{N}(e, \frac{e(1-e)}{N})$, i.e. set $Z = \frac{F-\mu}{\sigma} = \frac{F-e}{\sqrt{e(1-e)/N}}$ then of course $Z \sim \mathcal{N}(0,1)$. So $F = Z\sqrt{e(1-e)/N} + e$

  and

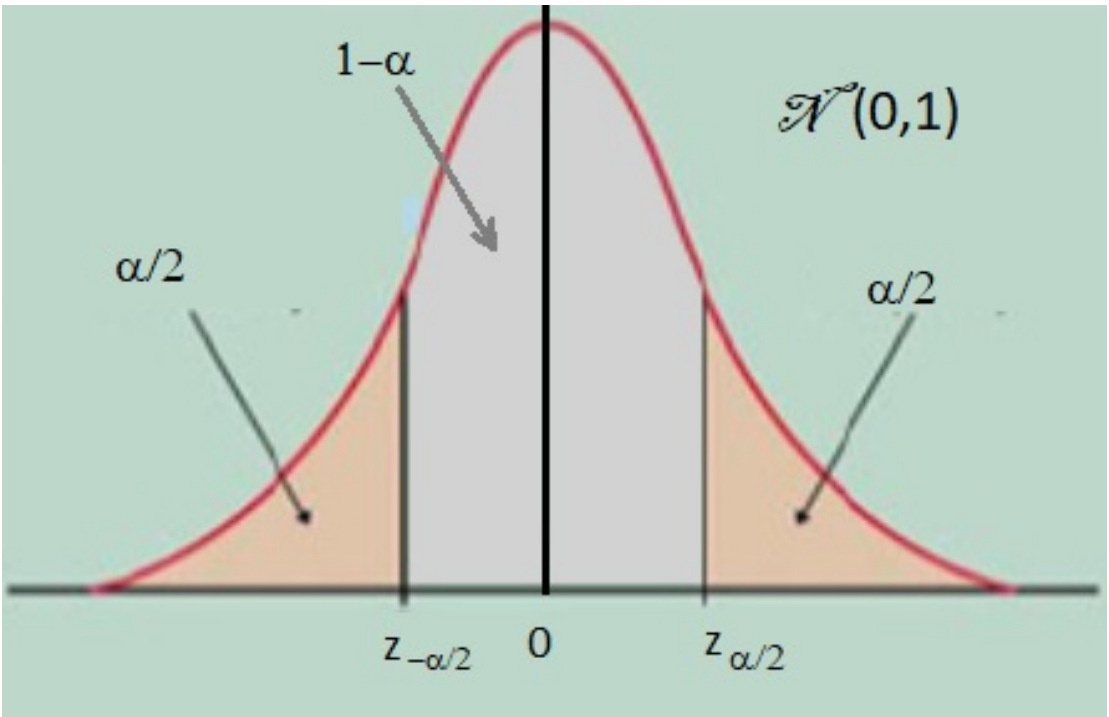
$$\begin{aligned}
\Pr\left(Z \leq \frac{r-e}{\sqrt{e(1-e)/N}}\right) &= \int_{\infty}^{\frac{r-e}{\sqrt{e(1-e)/N}}} \frac{1}{\sqrt{2\pi}} e^{-\frac{Z^2}{2}} dZ \\
&= \int_{\infty}^{r} \frac{1}{\sqrt{2\pi}} e^{-\frac{\left(\frac{F-e}{\sqrt{e(1-e)/N}}\right)^2}{2}} d\left(\frac{F-e}{\sqrt{e(1-e)/N}}\right) \\
&= \int_{\infty}^{r} \frac{1}{\sqrt{2\pi e(1-e)/N}} e^{-\frac{(F-e)^2}{2e(1-e)/N}} dF \\
&= \Pr(F \leq r)
\end{aligned}$$

- Since $\mathcal{N}(0,1)$ is tabulated/easily computed in $\mathbf{R}$ for any $1-\alpha$ we can find d such that $\Pr(Z \leq d) = \Pr(F \leq e + d\sqrt{e(1-e)/N}) = 1-\alpha$. But given F this only bounds e from below. What do we do?

one-Node Tree Error Estimation cont.

Normal distribution is symmetric, so we can lower and upper bound e by limiting Z (and consequently F) to an interval around mean cutting off tails:

$$\Pr\left(z_{-\frac{\alpha}{2}} \leq Z \leq z_{\frac{\alpha}{2}}\right) = \Pr\left(z_{-\frac{\alpha}{2}} \leq \frac{F-e}{\sqrt{e(1-e)/N}} \leq z_{\frac{\alpha}{2}}\right) = 1-\alpha$$



one-Node Tree Error Estimation cont.

Normal distribution is symmetric, so we can lower and upper bound e by limiting Z (and consequently F) to an interval around mean cutting off tails:

$$\Pr\left(z_{-\frac{\alpha}{2}} \leq Z \leq z_{\frac{\alpha}{2}}\right) = \Pr\left(z_{-\frac{\alpha}{2}} \leq \frac{F-e}{\sqrt{e(1-e)/N}} \leq z_{\frac{\alpha}{2}}\right) = 1-\alpha$$

So with probability $1-\alpha/2$ we have $z_{-\frac{\alpha}{2}}\sqrt{e(1-e)/N} \leq F-e$ so squaring both sides and subtracting left side we get inequality

$$F^2 - 2eF + e^2 - z^2_{-\frac{\alpha}{2}}\frac{e(1-e)}{N} = e^2\left(\frac{z^2_{-\frac{\alpha}{2}}}{N}+1\right) - e\left(\frac{z^2_{-\frac{\alpha}{2}}}{N}+2F\right) + F^2 \geq 0$$

Solving quadratic equation we obtain the only positive root:

$$e(F,N,\alpha) \leq \frac{F + \frac{z^2_{-\frac{\alpha}{2}}}{2N} + \left|z_{-\frac{\alpha}{2}}\right|\sqrt{\frac{F(1-F)}{N} + \frac{z^2_{-\frac{\alpha}{2}}}{4N^2}}}{\frac{z^2_{-\frac{\alpha}{2}}}{N}+1}$$

Example of One-node Tree Error Bounding

Let training data set for a single node classifier contain 200 data points with majority of 150 belonging to class 0 and 50 to class 1. So 50 data points are misclassified.
So $F = 50/200 = 1/4$, $N = 200$ what is error bound at confidence level 0.05?

Example of One-node Tree Error Bounding

Let training data set for a single node classifier contain 200 data points with majority of 150 belonging to class 0 and 50 to class 1. So 50 data points are misclassified.
So $F = 50/200 = 1/4$, $N = 200$ what is error bound at confidence level 0.05?

z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
-3.4	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0002
-3.3	0.0005	0.0005	0.0005	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0003
-3.2	0.0007	0.0007	0.0006	0.0006	0.0006	0.0006	0.0006	0.0005	0.0005	0.0005
-3.1	0.0010	0.0009	0.0009	0.0009	0.0008	0.0008	0.0008	0.0008	0.0007	0.0007
-3.0	0.0013	0.0013	0.0013	0.0012	0.0012	0.0011	0.0011	0.0011	0.0010	0.0010
-2.9	0.0019	0.0018	0.0018	0.0017	0.0016	0.0016	0.0015	0.0015	0.0014	0.0014
-2.8	0.0026	0.0025	0.0024	0.0023	0.0023	0.0022	0.0021	0.0021	0.0020	0.0019
-2.7	0.0035	0.0034	0.0033	0.0032	0.0031	0.0030	0.0029	0.0028	0.0027	0.0026
-2.6	0.0047	0.0045	0.0044	0.0043	0.0041	0.0040	0.0039	0.0038	0.0037	0.0036
-2.5	0.0062	0.0060	0.0059	0.0057	0.0055	0.0054	0.0052	0.0051	0.0049	0.0048
-2.4	0.0082	0.0080	0.0078	0.0075	0.0073	0.0071	0.0069	0.0068	0.0066	0.0064
-2.3	0.0107	0.0104	0.0102	0.0099	0.0096	0.0094	0.0091	0.0089	0.0087	0.0084
-2.2	0.0139	0.0136	0.0132	0.0129	0.0125	0.0122	0.0119	0.0116	0.0113	0.0110
-2.1	0.0179	0.0174	0.0170	0.0166	0.0162	0.0158	0.0154	0.0150	0.0146	0.0143
-2.0	0.0228	0.0222	0.0217	0.0212	0.0207	0.0202	0.0197	0.0192	0.0188	0.0183
-1.9	0.0287	0.0281	0.0274	0.0268	0.0262	0.0256	0.0250	0.0244	0.0239	0.0233
-1.8	0.0359	0.0351	0.0344	0.0336	0.0329	0.0322	0.0314	0.0307	0.0301	0.0294
-1.7	0.0446	0.0436	0.0427	0.0418	0.0409	0.0401	0.0392	0.0384	0.0375	0.0367
-1.6	0.0548	0.0537	0.0526	0.0516	0.0505	0.0495	0.0485	0.0475	0.0465	0.0455
-1.5	0.0668	0.0655	0.0643	0.0630	0.0618	0.0606	0.0594	0.0582	0.0571	0.0559
-1.4	0.0808	0.0793	0.0778	0.0764	0.0749	0.0735	0.0721	0.0708	0.0694	0.0681
-1.3	0.0968	0.0951	0.0934	0.0918	0.0901	0.0885	0.0869	0.0853	0.0838	0.0823
-1.2	0.1151	0.1131	0.1112	0.1093	0.1075	0.1056	0.1038	0.1020	0.1003	0.0985
-1.1	0.1357	0.1335	0.1314	0.1292	0.1271	0.1251	0.1230	0.1210	0.1190	0.1170
-1.0	0.1587	0.1562	0.1539	0.1515	0.1492	0.1469	0.1446	0.1423	0.1401	0.1379
-0.9	0.1841	0.1814	0.1788	0.1762	0.1736	0.1711	0.1685	0.1660	0.1635	0.1611
-0.8	0.2119	0.2090	0.2061	0.2033	0.2005	0.1977	0.1949	0.1922	0.1894	0.1867
-0.7	0.2420	0.2389	0.2358	0.2327	0.2296	0.2266	0.2236	0.2206	0.2177	0.2148
-0.6	0.2743	0.2709	0.2676	0.2643	0.2611	0.2578	0.2546	0.2514	0.2483	0.2451
-0.5	0.3085	0.3050	0.3015	0.2981	0.2946	0.2912	0.2877	0.2843	0.2810	0.2776
-0.4	0.3446	0.3409	0.3372	0.3336	0.3300	0.3264	0.3228	0.3192	0.3156	0.3121
-0.3	0.3821	0.3783	0.3745	0.3707	0.3669	0.3632	0.3594	0.3557	0.3520	0.3483
-0.2	0.4207	0.4168	0.4129	0.4090	0.4052	0.4013	0.3974	0.3936	0.3897	0.3859
-0.1	0.4602	0.4562	0.4522	0.4483	0.4443	0.4404	0.4364	0.4325	0.4286	0.4247
-0.0	0.5000	0.4960	0.4920	0.4880	0.4840	0.4801	0.4761	0.4721	0.4681	0.4641

Example of One-node Tree Error Bounding

Let training data set for a single node classifier contain 200 data points with majority of 150 belonging to class 0 and 50 to class 1. So 50 data points are misclassified.
So $F = 50/200 = 1/4$, $N = 200$ what is error bound at confidence level 0.05?

If $\alpha = 0.05$ then $\alpha/2 = 0.025$. By symmetry $\left|z_{-\frac{\alpha}{2}}\right| = z_{\frac{\alpha}{2}}$ so we can use upper tail. The get $z_{0.025}$ either from tables or from $\mathbf{R}$ using
`qnorm(.025,lower.tail=FALSE)` which gives $z_{0.025} = 1.96$

$$\begin{aligned} e(1/4, 200, 0.05) &\leq \frac{0.25 + \frac{1.96^2}{2\times 200} + 1.96\sqrt{\frac{1/4\times 3/4}{200} + \frac{1.96^2}{4\times 200^2}}}{\frac{1.96^2}{200} + 1} \\ &= 0.314 \end{aligned}$$

Error Estimation of 2-level Binary DTree Model

- The root of 2-level binary tree splits on $X < a$ for some feature X. As X is a random, the data is distributed between left node l and and right node r with (unknown to us) probability $p_l = p$ and $p_r = 1 - p$, i.e. we have Bernoulli trial $\mathscr{B}(p, 1-p)$. .
- Let e_l and e_r be error probabilities in l and r (on all data). The error event we are after is total error $e = (e_l \cap X < a) \cup (e_r \cup X \geq a)$. We can bound on errors e_L and e_R for leaf l (or for r) using single-node-tree model for subset of data with $X < a$ (resp. $X \geq a$). But errors e_L,e_R are conditional probabilities $e_L = \Pr(e_l | X < a)$ and $e_R = \Pr(e_r | X \geq a)$, so $e = e_L \times \Pr(X < a) + e_R \Pr(X \geq a)$. Thus if we bound the routing probability for the leaf with max-error, then we can compute the bound for e
- Let $e_L > e_R$ (flip L and R otherwise). In the sample $S = \{x_1, \ldots, x_N\}$ repeat the construction for bounding random variable p using $f = N_L/|S|$ where N_L counts the number of data points directed to left child in S. Assuming $f \leq 1/2$ bound p with given confidence level α using the same formula as for bounding node error. If $f \geq 1/2$ then lower bound on routing to the right probability $1 - p$ is needed to get upper bound on p. Since $z_{-\alpha} = -z_\alpha$ we can use the same formula to estimate lower bound g on $N_R/|S|$,so $p \leq 1 - g$. The obtained bound is meaningless (i.e. $p \lesssim 1$), so most of the time in practice either empiric frequency f is used in palce of p or the robust estimate $e = \max\{e_L, e_R\}$ is taken.

Error Estimation of 2-level Binary DTree Model

- Let e_l and e_r be error probabilities in l and r (on all data). The error event we are after is total error $e = (e_l \cap X < a) \cup (e_r \cup X \geq a)$. We can bound on errors e_L and e_R for leaf l (or for r) using single-node-tree model for subset of data with $X < a$ (resp. $X \geq a$). But errors e_L,e_R are conditional probabilities $e_L = \Pr(e_l | X < a)$ and $e_R = \Pr(e_r | X \geq a)$, so $e = e_L \times \Pr(X < a) + e_R \Pr(X \geq a)$. Thus if we bound the routing probability for the leaf with max-error, then we can compute the bound for e
- Let $e_L > e_R$ (flip L and R otherwise). In the sample $S = \{x_1, \ldots, x_N\}$ repeat the construction for bounding random variable p using $f = N_L/|S|$ where N_L counts the number of data points directed to left child in S. Assuming $f \leq 1/2$ bound p with given confidence level α using the same formula as for bounding node error.
- If we follow through with estimation of p, what is the confidence level of our combined error estimate? we need to guarantee the event $e_L = (e | X < a)$ because it contributes more to an error. Let α_1 be confidence level (probability) for event $e \leq b_1$ and α_2 be confidence level for the event $p \leq b_2$. The assuming the events are independent final confidence level is $\alpha = 1 - (1 - \alpha_1)(1 - \alpha_2)$

Example of 2-level Binary DTree Error Estimation

Let training data set for a 2-level binary tree classifier contain 200 data points of which 140 data points are classified in left node and 60 data points are classified in right node. In the left node 140 data points are classified correctly and 20 data points are misclassified. in the right node 30 data points are classified correctly and 10 data points are misclassified. We want to estimate the error at 0.05 confidence level

- so $N_L = 160$, $F_L = 20/160 = 1/8$, and $N_R = 40$, $F_R = 10/40 = 1/4$. Routing frequencies are $f_L = 160/200 = 4/5$ and $f_R = 40/200 = 1/5$
- To obtain confidence level 0.05 for combined error we'd like the same confidence levels for errors at the leafs and for probability of node routing we need to have $1 - 0.05 = (1 - \alpha)^2$ or $\alpha = 1 - \sqrt{0.95} = 0.0253$ so $\alpha/2 = 0.01265$. The latter gives $Z_{0.01265} = 2.237$

Example of 2-level Binary DTree Error Estimation

- so $N_L = 160$, $F_L = 20/160 = 1/8$, and $N_R = 40$, $F_R = 10/40 = 1/4$. Routing frequencies are $f_L = 160/200 = 4/5$ and $f_R = 40/200 = 1/5$
- To obtain confidence level 0.05 for combined error we'd like the same confidence levels for errors at the leafs and for probability of node routing we need to have $1 - 0.05 = (1-\alpha)^2$ or $\alpha = 1 - \sqrt{0.95} = 0.0253$ so $\alpha/2 = 0.01265$. The latter gives $Z_{0.01265} = 2.237$
- Then

$$\begin{aligned} e_L(1/8, 160, 0.0253) &\leq \frac{0.125 + \frac{2.237^2}{2\times 160} + 2.237\sqrt{\frac{1/8\times 7/8}{160} + \frac{2.237^2}{4\times 160^2}}}{\frac{2.237^2}{160}+1} \\ &\leq 0.195 \end{aligned}$$

and

$$\begin{aligned} e_R(1/4, 40, 0.0253) &\leq \frac{0.25 + \frac{2.237^2}{2\times 40} + 2.237\sqrt{\frac{1/4\times 3/4}{40} + \frac{2.237^2}{4\times 40^2}}}{\frac{2.237^2}{40}+1} \\ &\leq 0.425 \end{aligned}$$

Example of 2-level Binary DTree Error Estimation

- so $N_L = 160$, $F_L = 20/160 = 1/8$, and $N_R = 40$, $F_R = 10/40 = 1/4$. Routing frequencies are $f_L = 160/200 = 4/5$ and $f_R = 40/200 = 1/5$
- To obtain confidence level 0.05 for combined error we'd like the same confidence levels for errors at the leafs and for probability of node routing we need to have $1 - 0.05 = (1-\alpha)^2$ or $\alpha = 1 - \sqrt{0.95} = 0.0253$ so $\alpha/2 = 0.01265$. The latter gives $Z_{0.01265} = 2.237$
- then $e_L(1/8, 160, 0.0253) \leq 0.195$ and $e_R(1/4, 40, 0.0253) \leq 0.425$
- Since e_R bound is higher we need to bound routing probability to the right p_R

$$\begin{aligned} p_R(1/5, 40, 0.0253) &\leq \frac{0.20 + \frac{2.237^2}{2\times 40} + 2.237\sqrt{\frac{1/5\times 4/5}{40} + \frac{2.237^2}{4\times 40^2}}}{\frac{2.237^2}{40} + 1} \\ &\leq 0.371 \end{aligned}$$

- The probability p_L is then at least $1 - 0.371 = 0.629$. Then

$$\begin{aligned} e &\leq e_R(1/4, 40, 0.0253)p_R(1/5, 40, 0.0253) + e_L(1 - p_R(1/5, 40, 0.0253)) \\ &\leq 0.425 \times 0.371 + 0.195 \times 0.629 = 0.28 \end{aligned}$$

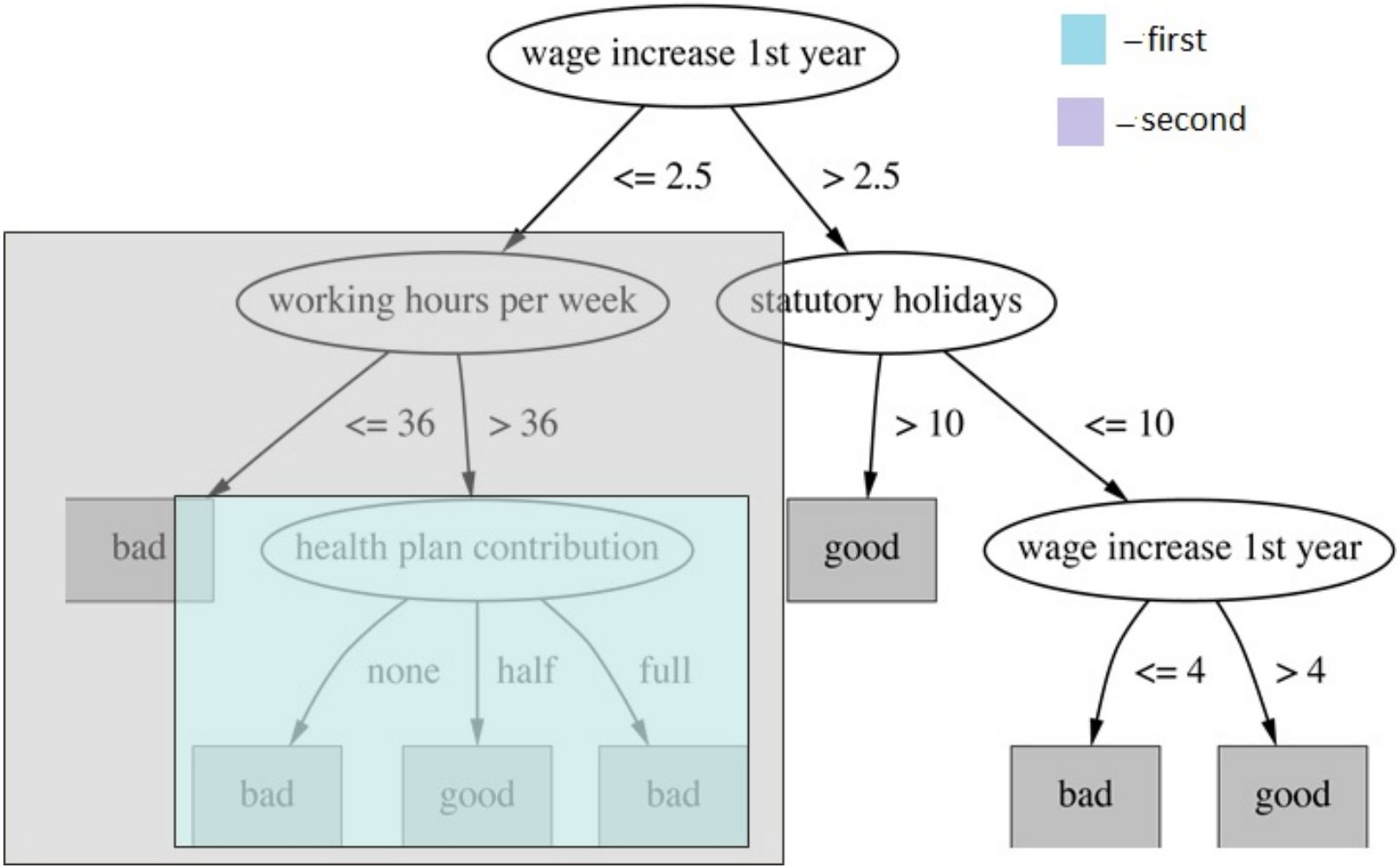
Lecture Overview

DFS Error-based Post-Pruning

- Bounding of error in multiway DTrees is straightforward. So for any 2-level Dtrees post-pruning the method is pretty simple:
  - Choose confidence level
  - Evaluate an error for singe node tree,
  - Evaluate an error for 2-level D-tree,
  - If the former is no greater than the latter then prune leaf nodes
- For an arbitrary level DTree apply the 2-level step recursively:
  - Walk the DTree in DFS manner
  - When in the leaf compute the error as in a singe node
  - When in a node all children of which are the leafs and all have been visited apply 2-level Dtree postpruning algorithms

DFS Error-based Post-Pruning

DFS Post-Pruning: *Bottom-up*! We consider replacing a tree only after considering all its subtrees

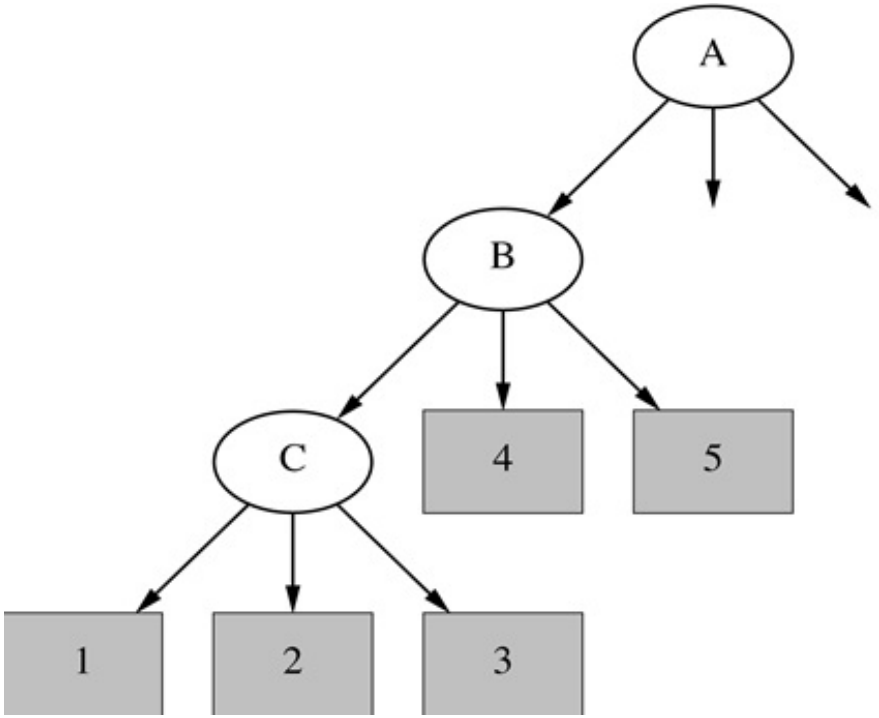


BFS Post-Pruning

- For an arbitrary level DTree apply the 2-level step walking the tree top down:
  - Walk the DTree in BFS manner
  - When in the node compute the error as in a singe node
  - When in a node all children of a node are already processed compute the error bound for a 2 level tree rooted in the node
  - If single node error is lower that 2 level Dtree error eliminate the split
  - Find the best split among splits used in each one of the children of this node, and redistribute instance from eliminated children
- This approach is also known as tree raising. Requires substantial recomputation of a tree and is much slower than DFS error-pruning.

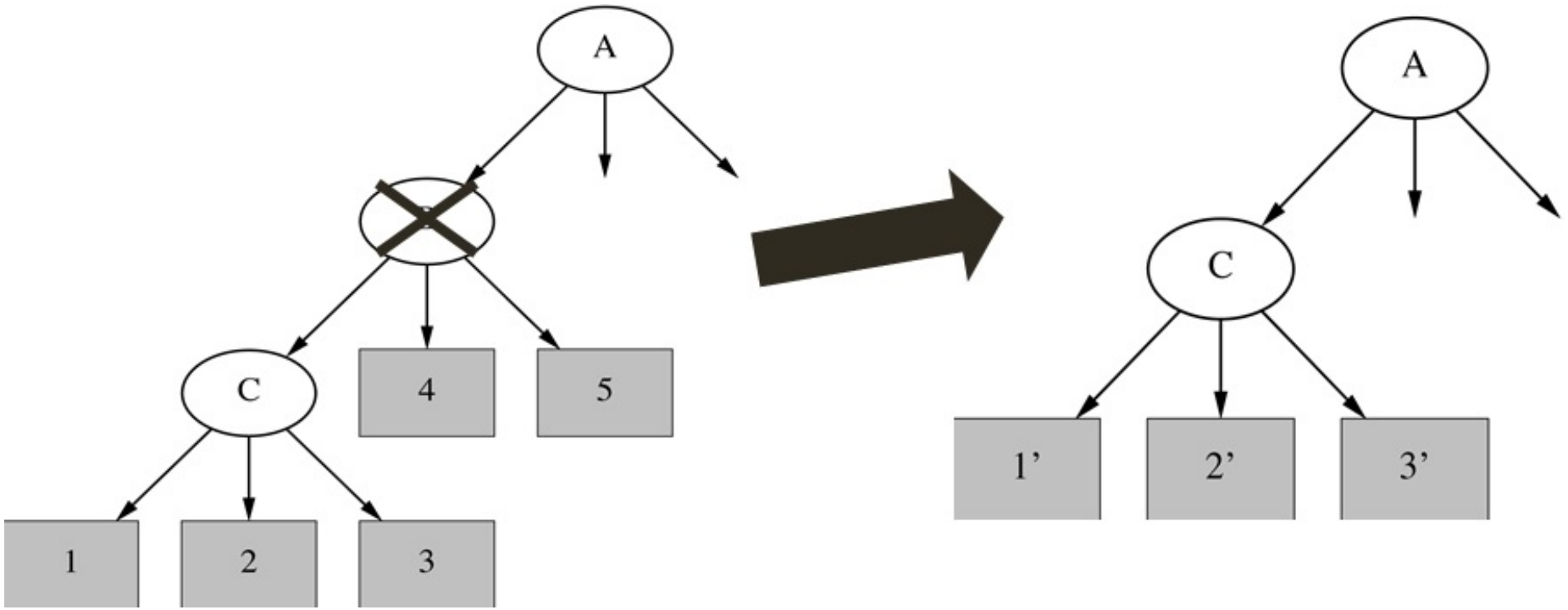
BFS Post-Pruning

BFS Post-Pruning: *Top-down*! Deleting node requires re-computation of a best split (only among children-used splits) and redistribution of data points to new nodes. It is essentially a DTree recomputation.
Is it worthwhile? practical answer is 'No'



BFS Post-Pruning

BFS Post-Pruning: *Top-down*! Deleting node requires re-computation of a best split (only among children-used splits) and redistribution of data points to new nodes. It is essentially a DTree recomputation.
Is it worthwhile? practical answer is 'No'



Pruning in C4.5

C4.5 algorithm (J48 in weka–$\mathbf{R}$) implements both types of pruning it is controlled by weka_control vector c parameter there specifies confidence level and S parameter BFS-pruning ($S = TRUE$) or DFS ($S = FALSE$). The following slides gives a code using extension of

C4.5 with different types of pruning for tree learning comparison (c5.0 implementation) :

- Without pruning - C5T tree
- With DFS pruning - C.5T.pop tree
- With DFS pruning and prepruning - C4.5T.prp tree
- in the latter two cases the confidence level is set to $c = 0.01$

Pruning in C4.5 - implementation c5.0.

```
library(sets); library(C50)
set.seed(500)
ind <- sample(2, nrow(iris), replace = TRUE, prob=c(2/3, 1/3))
irisTL<-iris[ind==1,]; # learning
irisTC<-iris[ind==2,] # testing
#no pruning at all min instances 5
C5T<-C5.0(Species ~ ., data=irisTL, control=C5.0Control(minCase = 5,
   noGlobalPruning = TRUE, earlyStopping = FALSE)); plot(C5T); C5T
#minimum instances is 5, no pruning no early stopping
pred.C5T <-predict(C5T,newdata=irisTC) # classify TC
acc.C5T <-sum(pred.C5T==irisTC$Species)/dim(irisTC)[1]; acc.C5T
# postpruning with confidence level 0.01
C5T.pop<-C5.0(Species ~ ., data=irisTL, control=C5.0Control(minCase = 5,
     noGlobalPruning = FALSE, earlyStopping = FALSE, CF=0.001));
plot(C5T.pop); pred.C5T.pop <-predict(C5T.pop,newdata=irisTC)
acc.C5T.pop <-sum(pred.C5T.pop==irisTC$Species)/dim(irisTC)[1];
acc.C5T.pop
#preprunning added same confidence level
C5T.prp<-C5.0(Species ~ ., data=irisTL, control=C5.0Control(minCase = 5,
  noGlobalPruning = FALSE, earlyStopping = TRUE, CF=0.01))
plot(C5T.prp); pred.C5T.prp <-predict(C5T.prp,newdata=irisTC)
acc.C5T.prp <-sum(pred.C5T.prp==irisTC$Species)/dim(irisTC)[1]
acc.C5T.prp
```