Random Forest

AbdElMoniem Bayoumi, PhD

Fall 2021

Uniform Sampling with Replacement

- Give a set of training examples $\boldsymbol{S}$

- Sample a subset $\boldsymbol{S_i}$ of examples

- **Uniform sampling:** selection probability of any example in $\boldsymbol{S}$ is $\frac{\mathbf{1}}{|\boldsymbol{S}|}$

- **Replacement:** a selected item can be reselected multiple times for the same subset

Random Forest

ehna el hadaf enna nkwn t Trees.
kol tree hya kawyea fl goz2 bta3ha.
b7es lama akhud ra2y kol el trees,
m3 b3d ytl3oly karar bnesba kbera s7.

- Given M training examples
- Uniformly sample T subsets with replacement
  - each of size M

el size bta3 kol el trees motsawy, 7agmohom M
fa keda hayb2a fe kol tree, fe samples mtkrra fe trees tanya.

b2dr atl3 el Size M, da 34an ana b3ml replacement, fa mmkn msln lw 3ndy 2 blue balls, ana mmkn wna bsample atl3 4 blue , 34an el 4 mrat el e5trthom, kano blue :)

- Build T decision trees with zero-training error
  - Tree for each training subset S_i
  - No pruning
- Take the average/votes of T trees
  - Subsets are different so no overfitting!
  - Can get certainty of your prediction, i.e., how many classifiers?

Random Forest

- Given D features, i.e., dimensions

- Building T decision trees:
  - Sample K features randomly $(K < D)$
  - Only split on these K random features
  - New K features are sampled for every single split

K is constant,
K can be sampled with replacement.

- Sampling features leads to different trees
  - Different mistakes by each tree
  - Mistakes averaged out → no overfitting

Random Forest

- Out of the box approach
  - No need to tune hyperparameters
  - No need to pre-process or scale inputs

- $K = \lceil \sqrt{D} \rceil$ → **round up!**

- Increase T as much as you can afford
  - Parallel processing

depending on the amount of resources that you have

two factors we have are
K -> etf2o enha ahsn haga el ceil(sqrt(D))
T -> kol ma tzwd kol ma yeb2a ahsn

Random Forest: Out-of-Bag Error

- Aka. out-of-bag estimate

data is splitted as :
1. Training -> bmrn beha el data.
2. Validation -> de el data elly b5ly el model bta3y yeshofha w a3adel el parameters bt3to w a3ml tuning lehom 34an a7asn el training.
3. Testing. -> to evaluate the accuracy.

tb ana keda msh fahem el fr2 ben el training wl validation...

- No need for training/validation split

el fekra bta3t el validation, enk bdl ma yeb2a 3ndk training w te3ml evaluation 3la el test bs, laa enta kol el bt3mlo enk btaa5ud goz2 mn el data te3ml beh training, w b3den te3ml evaluation bl validation of data, w b3den lw l2et el accuracy msh kwysa t3dl el hyperparameters b7es t7sn el accuracy bta3k baa.

- Can estimate test error directly from training set

kol element fl test, ana b5du a5leh yemshy 3la kol el

- Compute error for each training example
  - consider only trees that do not include that example in their training subset
  - approx. 60% of trees are considered

wna b3ml validation, msh b3ml consider lel trees elly el point de kant goz2 menha, lakn b5ud el trees el tanyen.

Random Forest: Out-of-Bag Error

- For each data point, average the loss of classifiers that do not have that data point,
  - we obtain an estimate of the true test error
  - without reducing the training set

$$E_{OFB} = \frac{1}{M}\sum_{i=1}^{M}\left[\frac{1}{z_i}\sum_{\substack{j \\ (x_i,y_i)\notin S_j}}^{T} loss\left(h_j(x_i), y_i\right)\right]$$

lw 3ndk msln 10 trees
w 100 point
kol point btb2a mawgoda fe 3dd mo3n mn el trees lakn msh kolohom

fa lw hana5ud awl no2ta msln hya kant fe 6 trees

htro7 t7sb el loss fe el 4 trees elly hya mkntsh fehom
w ne2sm 3la 10
(fa de keda el 1/z sigma loss)

$$z_i = \sum_{\substack{j \\ (x_i,y_i)\notin S_j}}^{T} 1$$

num of trees **NOT** trained on (x_i, y_i)

w htkrr baa nafs el 3amalya lkol el points w fl akher baa bt2sm 3la 3adad el samples el 3ndk

Random Forest

- Easy to use!
- Second best approach! → **Rule of thumb!**
- No need for training/validation split
- For regression, use regression trees

Random Forest

- Wisdom of the crowd!

- Improvement: prune the last split of trees (i.e., from bottom) if that improves E_{OFB}
  - Decrease size of trees
  - Decrease noise

- Not suitable for raw images

Acknowledgement

- These slides have been designed relying on materials of Victor Lavrenko and Kilian Weinberger