Machine Learning Lecture 7: Bias-Variance tradeoff; Stacking, Blending

Iurii Efimov

girafe ai

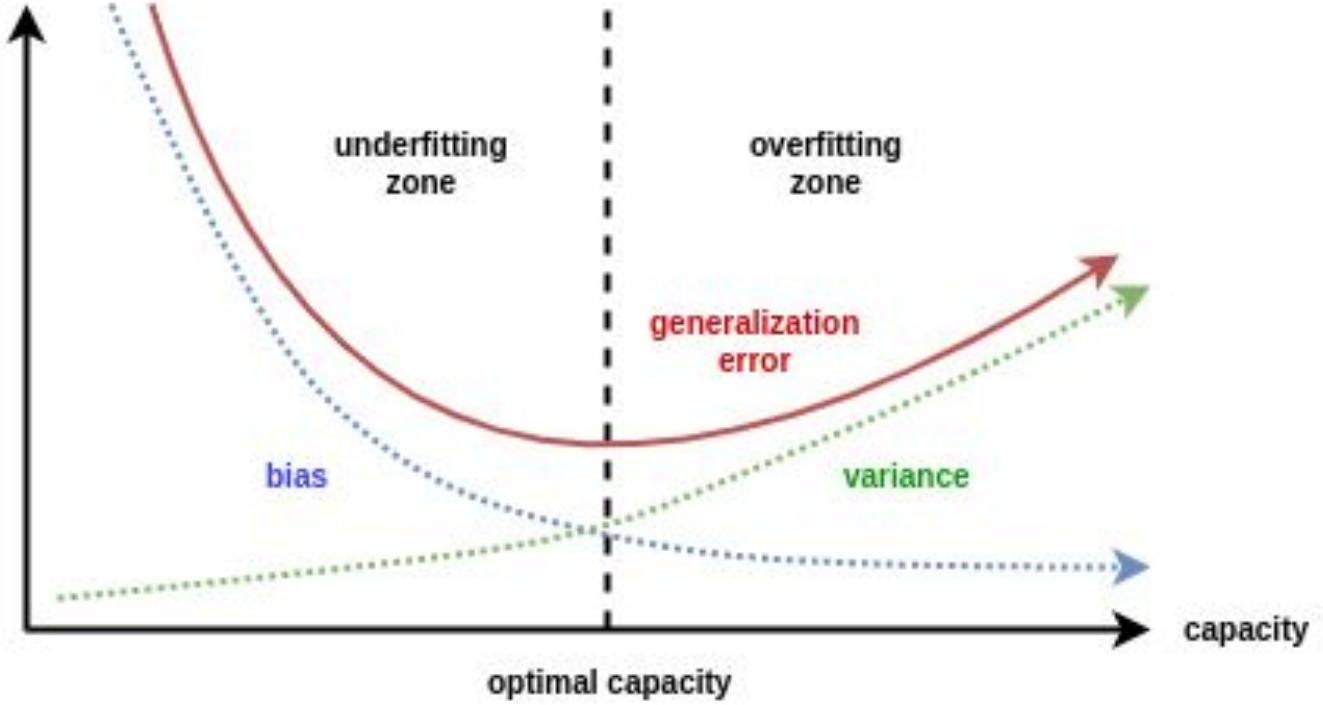
Outline

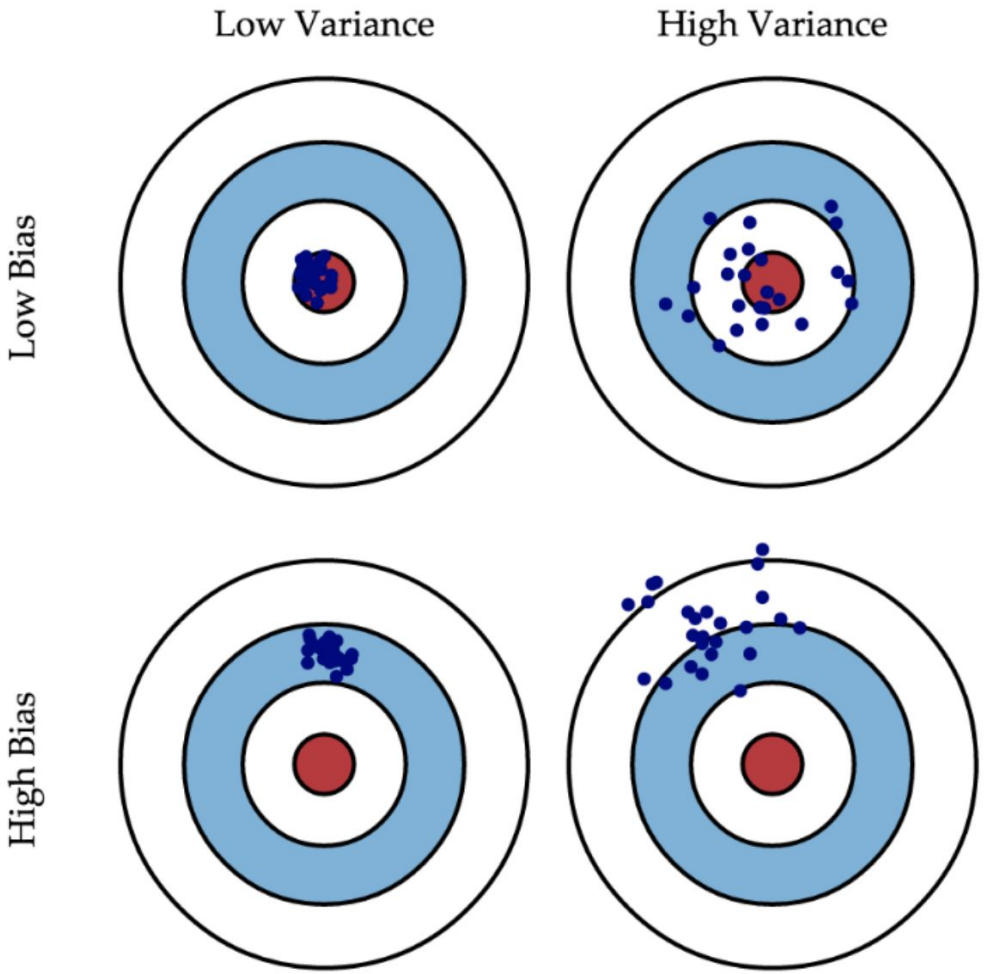
- Bias-Variance Tradeoff
- Blending
- Stacking

Bias-Variance tradeoff

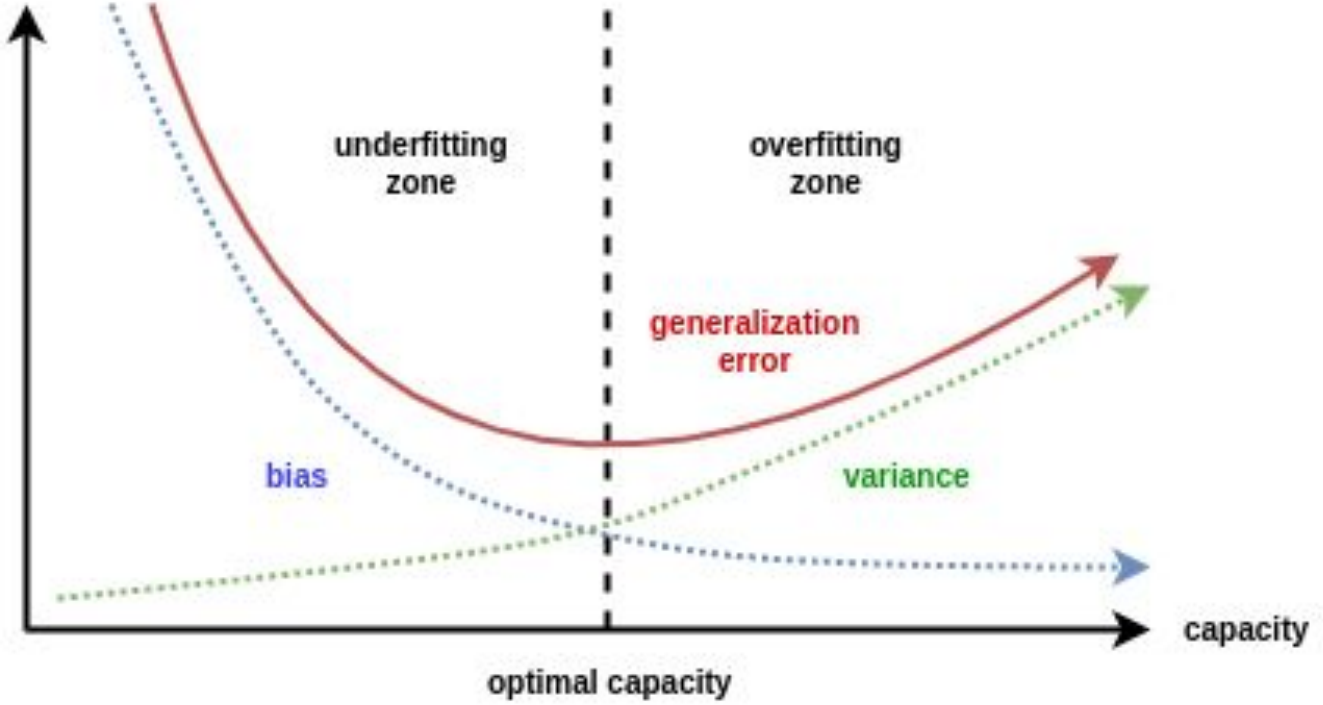
girafe
ai

Bias-variance tradeoff




Low Variance
High Variance
Low Bias
High Bias

Bias-variance tradeoff

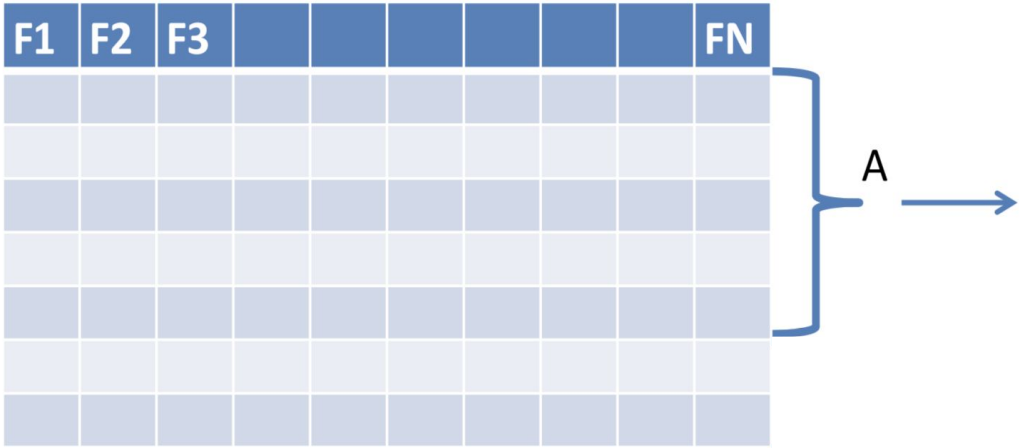


Stacking and blending

girafe
ai

Blending

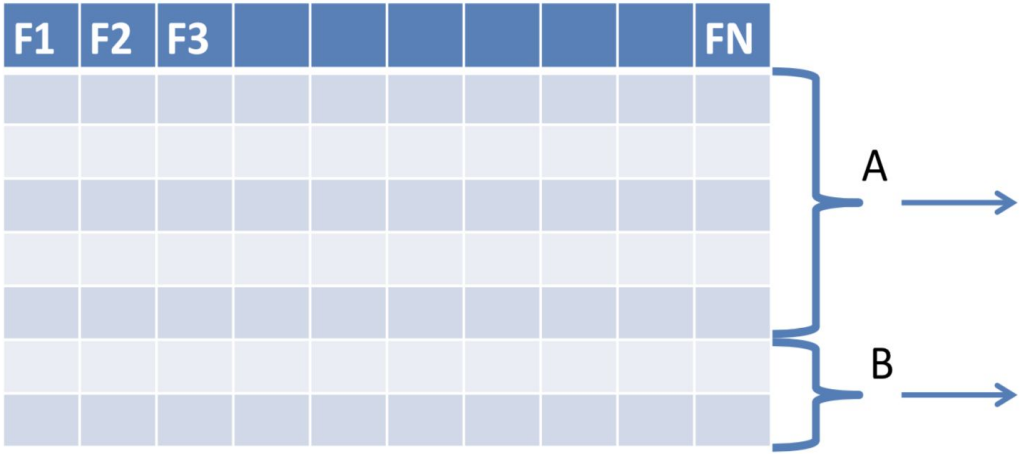
How to build an ensemble from different models?



Tune M basic algorithms

Blending

How to build an ensemble from different models?

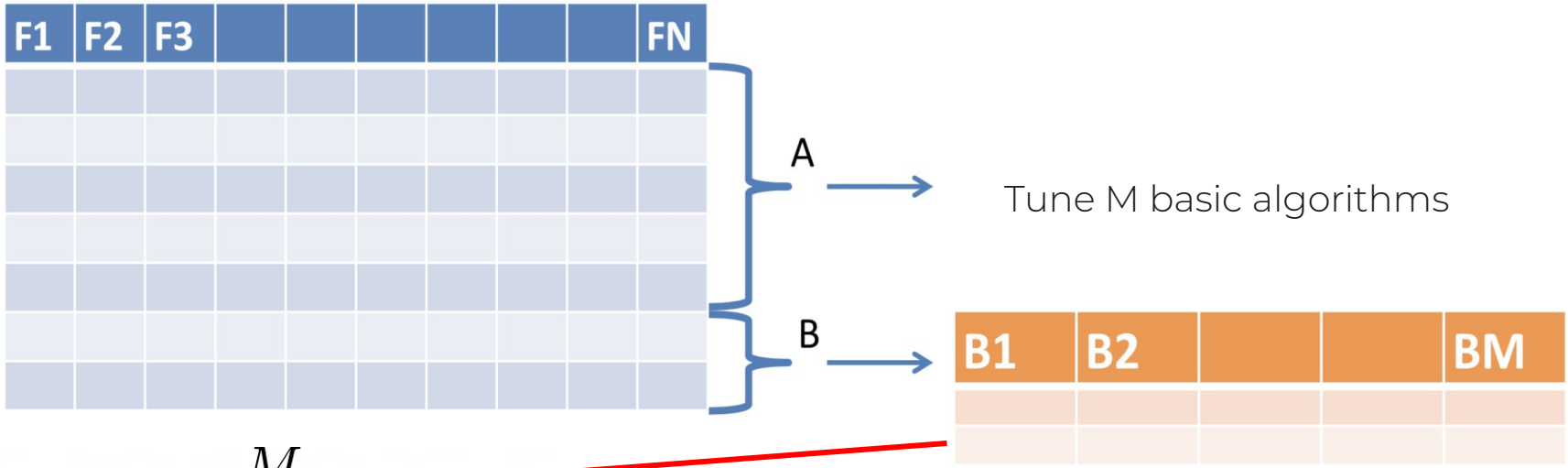


Tune M basic algorithms

Tune new model on the outputs of the basic algorithms

Blending

How to build an ensemble from different models?



$$\hat{f}(x) = \sum_{i=1}^{M} \rho_i f_i(x)$$

$$\sum_{i=1}^{M} \rho_i = 1, \quad \rho_i \in [0;1] \ \ \forall i$$

Blending

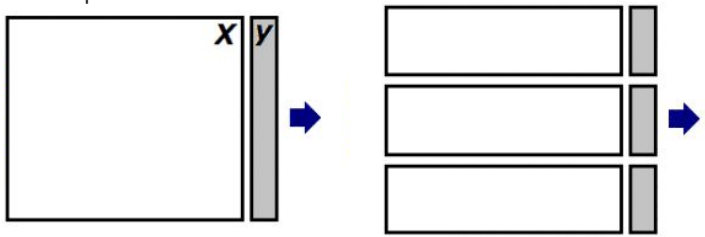
Just combine several strong/complex models.

$$\hat{f}(x) = \sum_{i=1}^{M} \rho_i f_i(x) \quad \sum_{i=1}^{M} \rho_i = 1, \quad \rho_i \in [0;1] \ \ \forall i$$

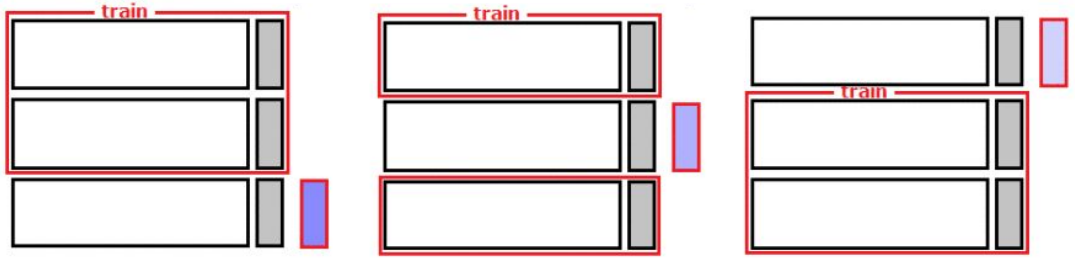
- Pros:
  - Simple and intuitive ensembling method.
  - Average several blendings to achieve better results.
- Cons:
  - Linear composition is not always enough.
  - Need to split the data. How to fix it?

Stacking

1. Split data into folds



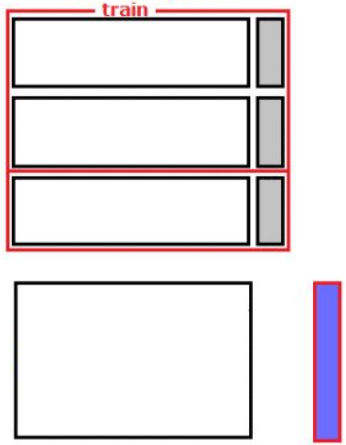
Fit using folds to get meta-features on train



2. Tune models on different groups of folds, predict on left out

3. Tune the new model on the “meta”-features

Fit using all data meta-features on test



Picture source: https://dyakonov.org/2017/03/10/стекинг-stacking-и-блендинг-blending/

Stacking

- Train base algorithm(s) on different groups of folds leaving one fold out.
- Predict the meta-features on the left-out fold and test data.
- Train the meta-algorithm on the meta-features representation of the train data.
- Use it on the meta-features representation of the test data.

Stacking

- Pros:
  - Powerful ensembling method, if you know how to use it
  - Quite popular in ML-competitions
  - One might perform stacking on the meta-features dataset as well
- Cons:
  - Meta-features on each fold are actually predicted by different models
    - However, regularization usually helps
  - Hard to explain your model behaviour

Stacking

Bonus:

Now you know how to stack XGBoost (or CatBoost/LightGBM)

Recap: ensembling methods

1. Bagging.
2. Random subspace method (RSM).
3. Bagging + RSM + Decision trees = Random Forest.
4. Gradient boosting.
5. Blending.
6. Stacking.

Great demo:
http://arogozhnikov.github.io/2016/06/24/gradient_boosting_explained.html

Revise

Thanks for attention!

Questions?