M10. Combined models and Ensemble methods

B Ravindran

PRML Jul-Aug 2021 (BR section)

Acknowledgment of Sources

- Slides based on content from related
  - Courses:
    - IITM – Profs. Arun/Harish/Chandra's PRML offerings (slides, quizzes, notes, etc.), Prof. Ravi's "Intro to ML" slides – cited respectively as [AR], [HR], [CC], [BR] in the bottom right of a slide.
    - India – NPTEL PR course by IISc Prof. PS. Sastry (slides, etc.) – cited as [PSS] in the bottom right of a slide.
  - Books:
    - PRML by Bishop. (content, figures, slides, etc.) – cited as [CMB]
    - Pattern Classification by Duda, Hart and Stork. (content, figures, etc.) – [DHS]
    - Mathematics for ML by Deisenroth, Faisal and Ong. (content, figures, etc.) – [DFO]
    - Information Theory, Inference and Learning Algorithms by David JC MacKay – [DJM]

Outline for Module M10

- M10. Combined models and Ensemble methods
  - **M10.0 Introduction/Motivation**
  - M10.1 Combined models
    - Conditional mixture models
    - Decision trees
  - M10.3 Ensemble methods
    - Parallel ensemble methods (bagging)
    - Sequential ensemble methods (boosting)
  - M10.4 Parting thoughts

Machine Learning in Practice

- Real world machine learning problems rarely have a unique and single best solution.
- Multiple thought processes and teams and approaches often yield equally valid but completely different solutions.
- The set of methods for combining many such solutions (classifiers, regressors etc.) into one solution are known as “Ensemble methods”

[Mostly based on Harish’s notes/slides; HR]

The Netflix Challenge

- Data set of ~100M star ratings that ~500K users gave to ~18K movies.
- Training data: *<user, movie, date of grade, grade>*
- The grand prize of $1,000,000, to be given to a team which beat Netflix's rating prediction algorithm by 10%

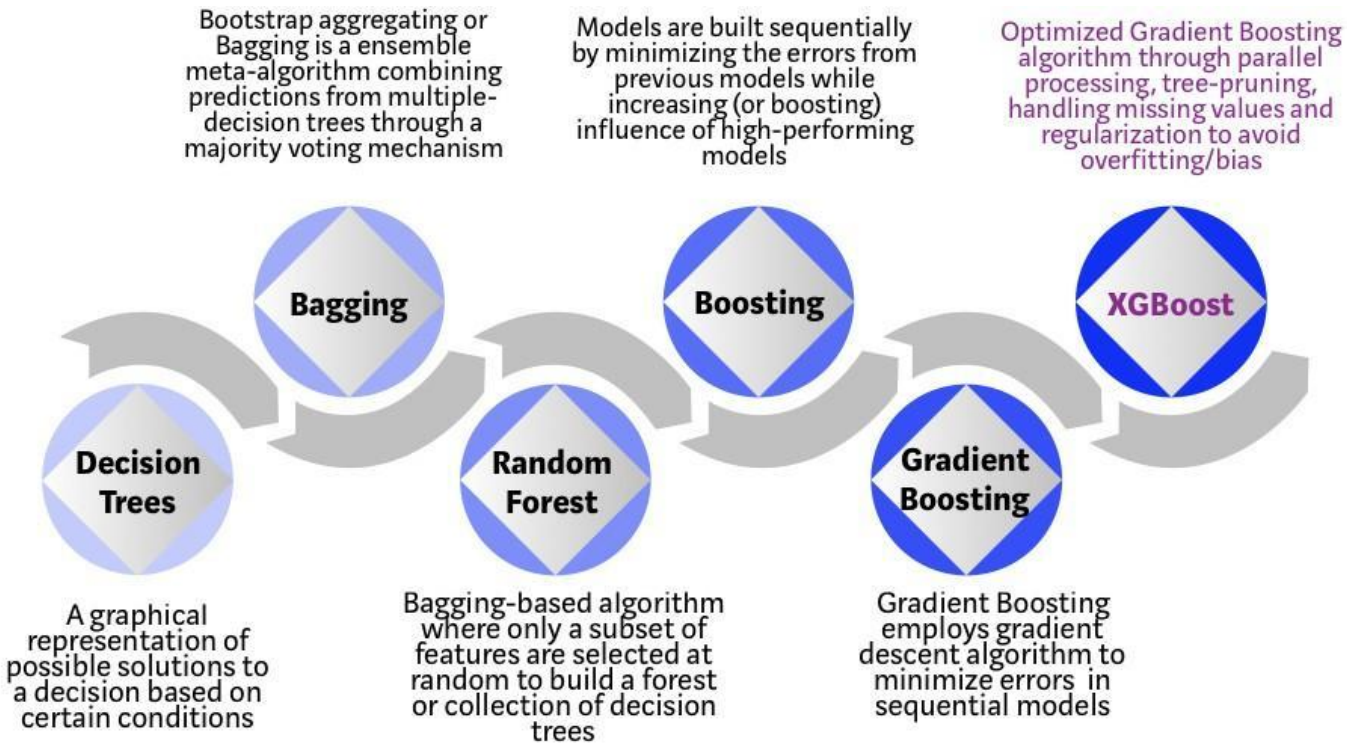
[HR]

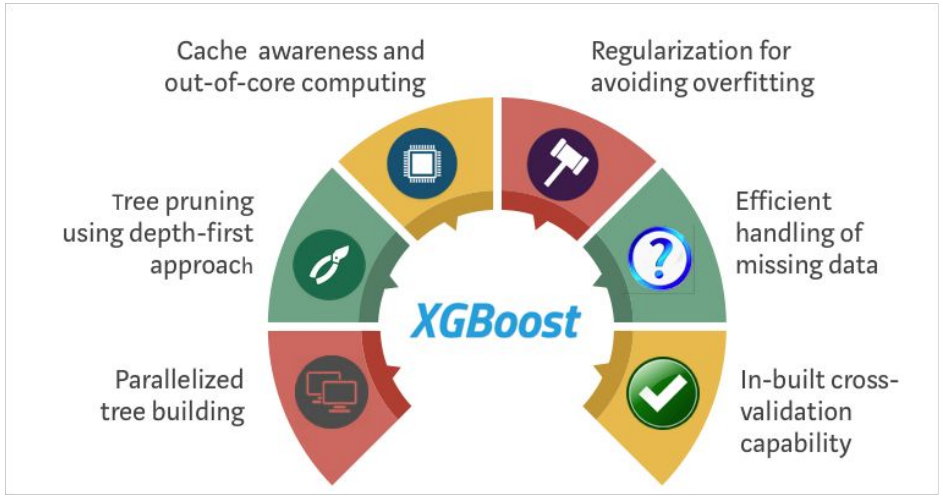
Netflix challenge: Winning Solution

- The winning team — "BellKor's Pragmatic Chaos" (itself a merger of several teams) combined a total of **107** separate prediction models!!
- The methods used various approaches — factor models, regression models, neighbourhood models, etc.
- Most other teams solutions also included large numbers of disparate models combined together.
- Ensemble methods are almost always the state of the art in any large scale Machine learning problem.

[HR]

“XGBoost Algorithm: Long May She Reign!”*





*by Vishal Morde [https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d]

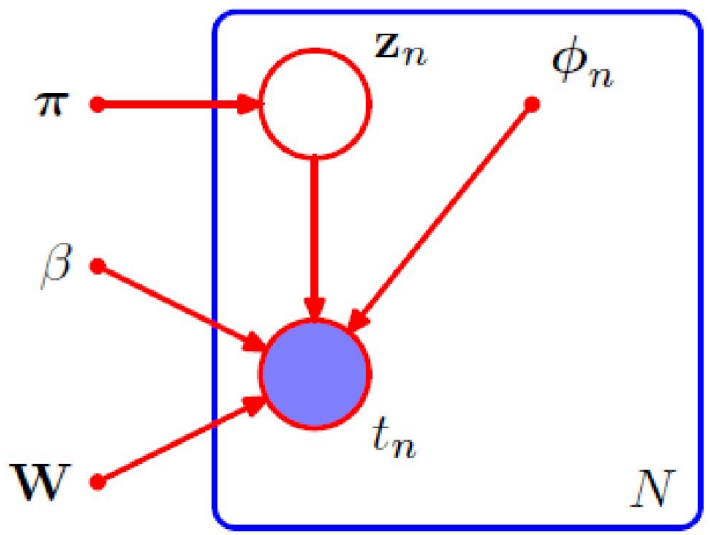
Outline for Module M10

- M10. Combined models and Ensemble methods
  - M10.0 Introduction/Motivation
  - **M10.1 Combined models**
    - **Conditional mixture models**
    - Decision trees
  - M10.3 Ensemble methods
    - Parallel ensemble methods (bagging)
    - Sequential ensemble methods (boosting)
  - M10.4 Parting thoughts

Conditional mixture models: mixture of linear regression models

$$p(t|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(t|\mathbf{w}_k^{\mathrm{T}}\boldsymbol{\phi}, \beta^{-1})$$

$$\gamma_{nk} = \mathbb{E}[z_{nk}] = p(k|\boldsymbol{\phi}_n, \boldsymbol{\theta}^{\mathrm{old}}) = \frac{\pi_k \mathcal{N}(t_n|\mathbf{w}_k^{\mathrm{T}}\boldsymbol{\phi}_n, \beta^{-1})}{\sum_j \pi_j \mathcal{N}(t_n|\mathbf{w}_j^{\mathrm{T}}\boldsymbol{\phi}_n, \beta^{-1})}.$$



[CMB]

MLE using EM algo.

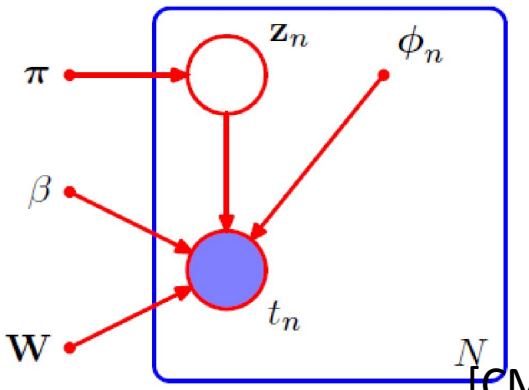
- Maximize (Marginal) Likelihood: $$\ln p(\mathbf{t}|\boldsymbol{\theta}) = \sum_{n=1}^{N} \ln \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(t_n|\mathbf{w}_k^{\mathrm{T}}\boldsymbol{\phi}_n, \beta^{-1}) \right)$$

- E-step: Compute responsibilities γ_{nk} .
- M-step: Maximize $$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\mathrm{old}}) = \mathbb{E}_{\mathbf{Z}} \left[\ln p(\mathbf{t}, \mathbf{Z}|\boldsymbol{\theta}) \right] = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \left\{ \ln \pi_k + \ln \mathcal{N}(t_n|\mathbf{w}_k^{\mathrm{T}}\boldsymbol{\phi}_n, \beta^{-1}) \right\}.$$

$$\pi_k = \frac{1}{N} \sum_{n=1}^{N} \gamma_{nk}.$$

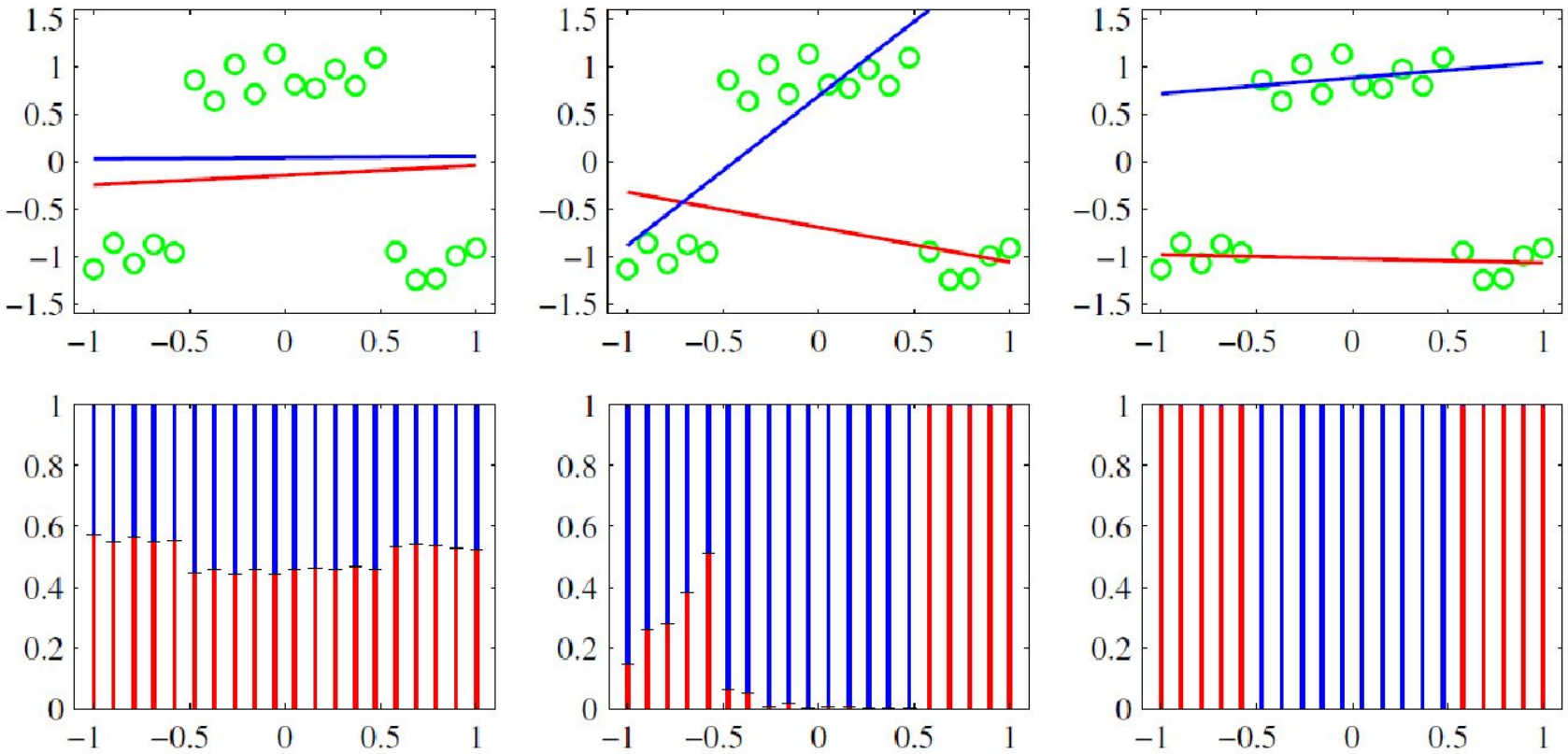
$$\mathbf{w}_k = \left(\boldsymbol{\Phi}^{\mathrm{T}} \mathbf{R}_k \boldsymbol{\Phi} \right)^{-1} \boldsymbol{\Phi}^{\mathrm{T}} \mathbf{R}_k \mathbf{t}. \qquad \mathbf{R}_k = \mathrm{diag}(\gamma_{nk})$$

$$\frac{1}{\beta} = \frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \left(t_n - \mathbf{w}_k^{\mathrm{T}} \boldsymbol{\phi}_n \right)^2.$$



[CMB]

Example: Mixtures of linear regression models



[CMB]

Other conditional mixture models

- Mixture of linear classifiers (logistic regression instead of linear regression models)

- Mixture of experts:
  - Key idea: allow mixing coefficients to be a function of the input x:

$$p(\mathbf{t}|\mathbf{x}) = \sum_{k=1}^{K} \pi_k(\mathbf{x}) p_k(\mathbf{t}|\mathbf{x}).$$

  - Compare with MDNs where all parameters can be input-dependent!

$$p(\mathbf{t}|\mathbf{x}) = \sum_{k=1}^{K} \pi_k(\mathbf{x}) \mathcal{N}\left(\mathbf{t}|\boldsymbol{\mu}_k(\mathbf{x}), \sigma_k^2(\mathbf{x})\right).$$

  - Hierarchical mixture of experts also possible, though we will look at **decision trees** as its hard (non-probab.) version of combining different models:

[CMB]

Outline for Module M10

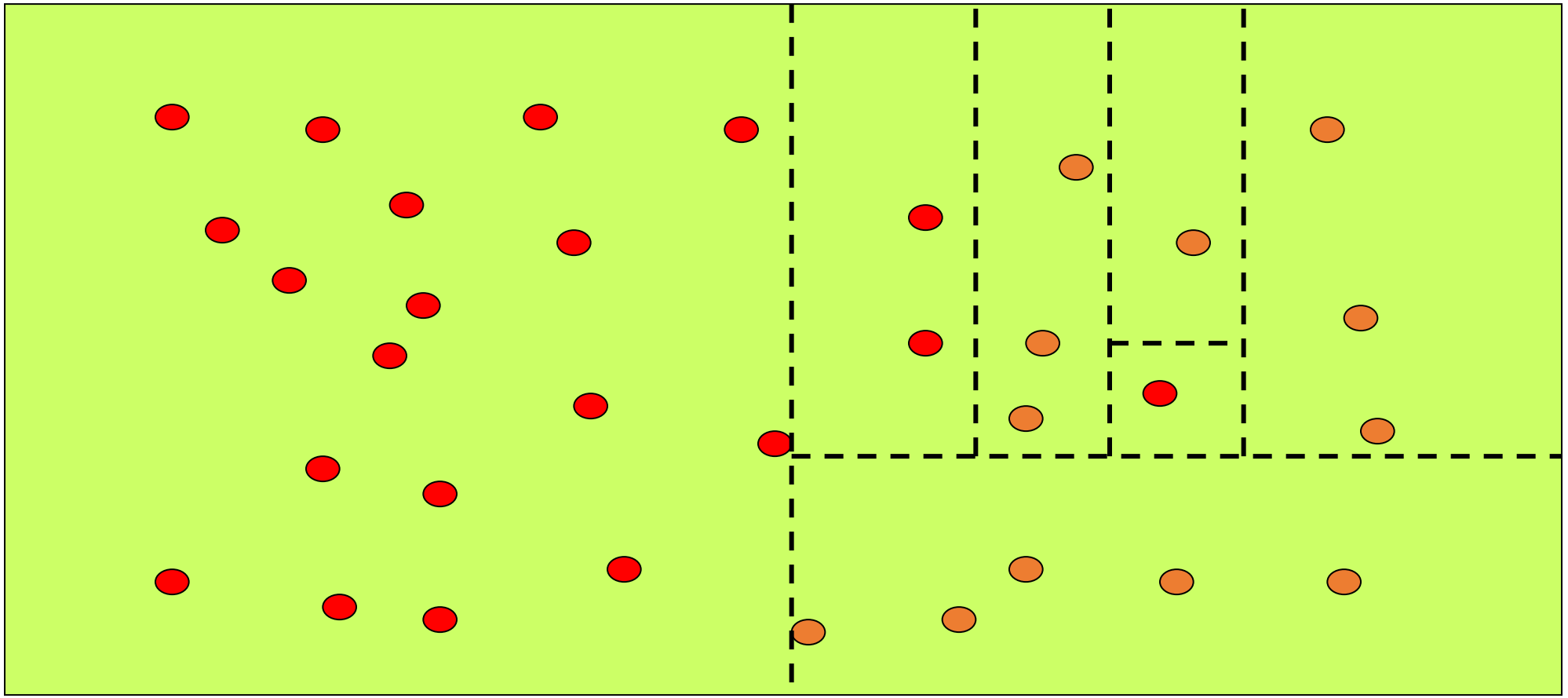
- M10. Combined models and Ensemble methods
  - M10.0 Introduction/Motivation
  - **M10.1 Combined models**
    - Conditional mixture models
    - **Decision trees**
  - M10.3 Ensemble methods
    - Parallel ensemble methods (bagging)
    - Sequential ensemble methods (boosting)
  - M10.4 Parting thoughts

A visual understanding of decision trees

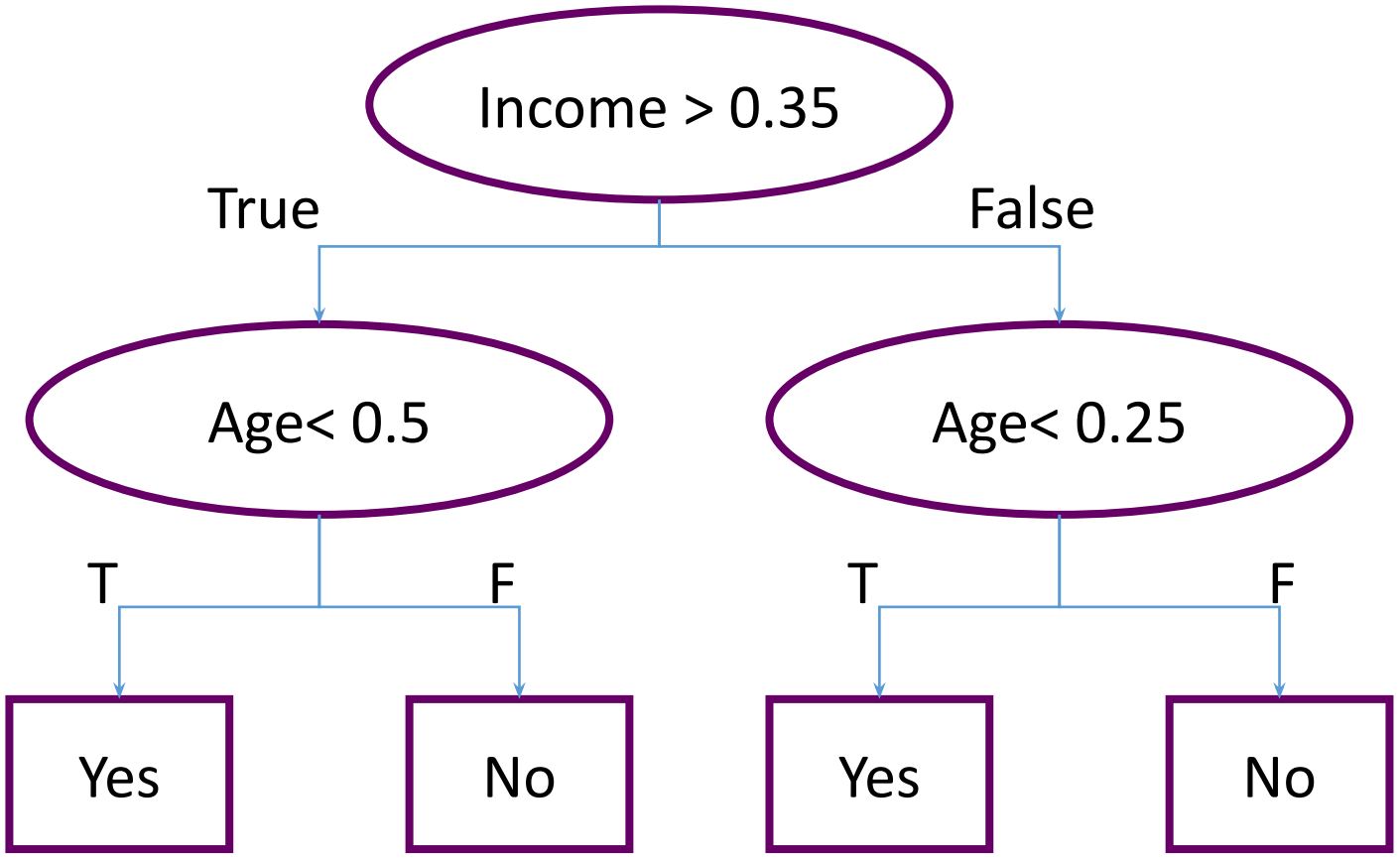
- http://www.r2d3.us/visual-intro-to-machine-learning-part-1/

Decision Trees

Decision boundaries



A Decision Tree

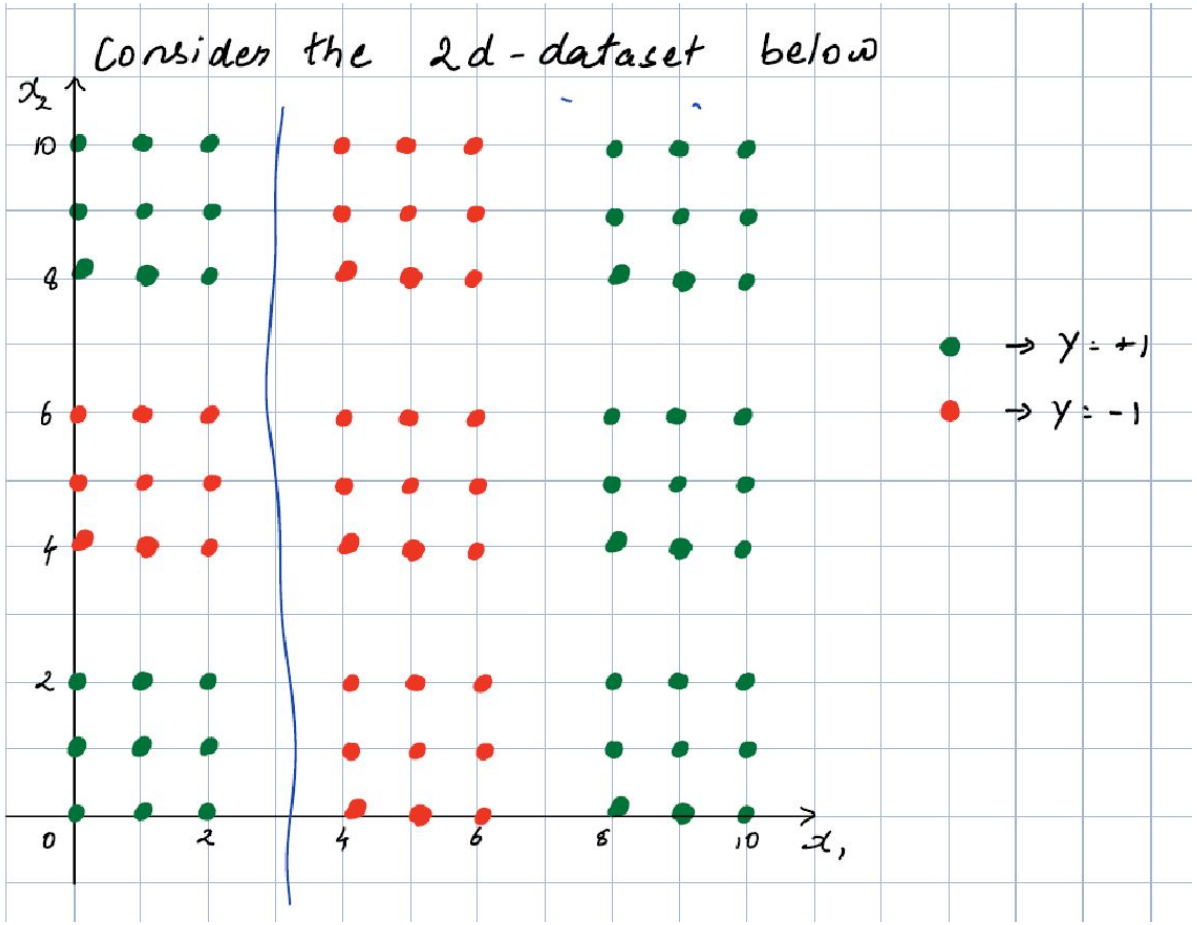



```
If Income > 0.35 and
   Age < 0.5
   then Buys = Yes
If Income > 0.35 and
   Age >= 0.5
   then Buys = No
If Income <= 0.35 and
   Age < 0.25
   then Buys = Yes
If Income <= 0.35 and
   Age >= 0.25
   then Buys = No
```

Constructing decision trees

- Strategy: top down
  Recursive *divide-and-conquer* fashion
  - First: select attribute for root node
    Create branch for each possible attribute value
  - Then: split instances into subsets
    One for each branch extending from the node
  - Finally: repeat recursively for each branch, using only instances that reach the branch
- Stop if all instances have the same class

Learning a decision tree – toy example



1.) What should the Root node be?

· Evaluate all classifiers of the form on "Entire" Training.

$$h(x) = \begin{cases} +1 & \text{if } x_1 \geq a \\ -1 & \text{if } x_1 < a \end{cases} \quad \text{and}$$

$$h(x) = \begin{cases} +1 & \text{if } x_2 \geq a \\ -1 & \text{if } x_2 < a \end{cases}$$

and their negations.

[HR]

Toy example – evaluating all "feature X threshold" combinations at the root node!

Accuracy

For simplicity we will evaluate only 4 such classifiers:

(a) $h(x) = +1$ if $x_1 \geq 7$
$= -1$ if $x_1 < 7$

Accuracy $= \frac{3+4}{9} = \frac{7}{9}$

(a') $h(x) = -1$ if $x_1 \geq 7$
$= +1$ $x_1 \leq 7$

Accuracy $= \frac{0+2}{9} = \frac{2}{9}$

Let L: $x_1 < 7$
R: $x_1 \geq 7$

Entropy

Define $H(p) = p \log \frac{1}{p} + (1-p) \log \left(\frac{1}{1-p}\right)$. (logarithm base is 2)

Avg. Entropy of split (a) $= P_L H_L + P_R H_R$.

P_L = Fraction of Points on the left. e.g. above $P_L = \frac{54}{81}$

$H_L = H(q_L)$ Where q_L = Fraction of Positive Points on the left. e.g above $q_L = \frac{18}{54}$

$\therefore$ Entropy of split (a) above

$= \frac{54}{81} H\left(\frac{18}{54}\right) + \frac{27}{81} H(1)$

$= \frac{2}{3} H\left(\frac{1}{3}\right) + \frac{1}{3} H(1) = \frac{2}{3} H\left(\frac{1}{3}\right)$

$= \frac{2}{3}\left(\frac{1}{3} \log 3 + \frac{2}{3} \log \frac{3}{2}\right) = \frac{2}{3}\left(\log 3 - \frac{2}{3}\right)$

[HR]

Exercise: try for 3 other splits (& their negations) to find the best root node split.

- E.g.,

(b) $h(x) = +1$ if $x_1 \geq 3$
$= -1$ if $x_1 < 3$

Accuracy $= \frac{3+1}{9} = \frac{4}{9}$

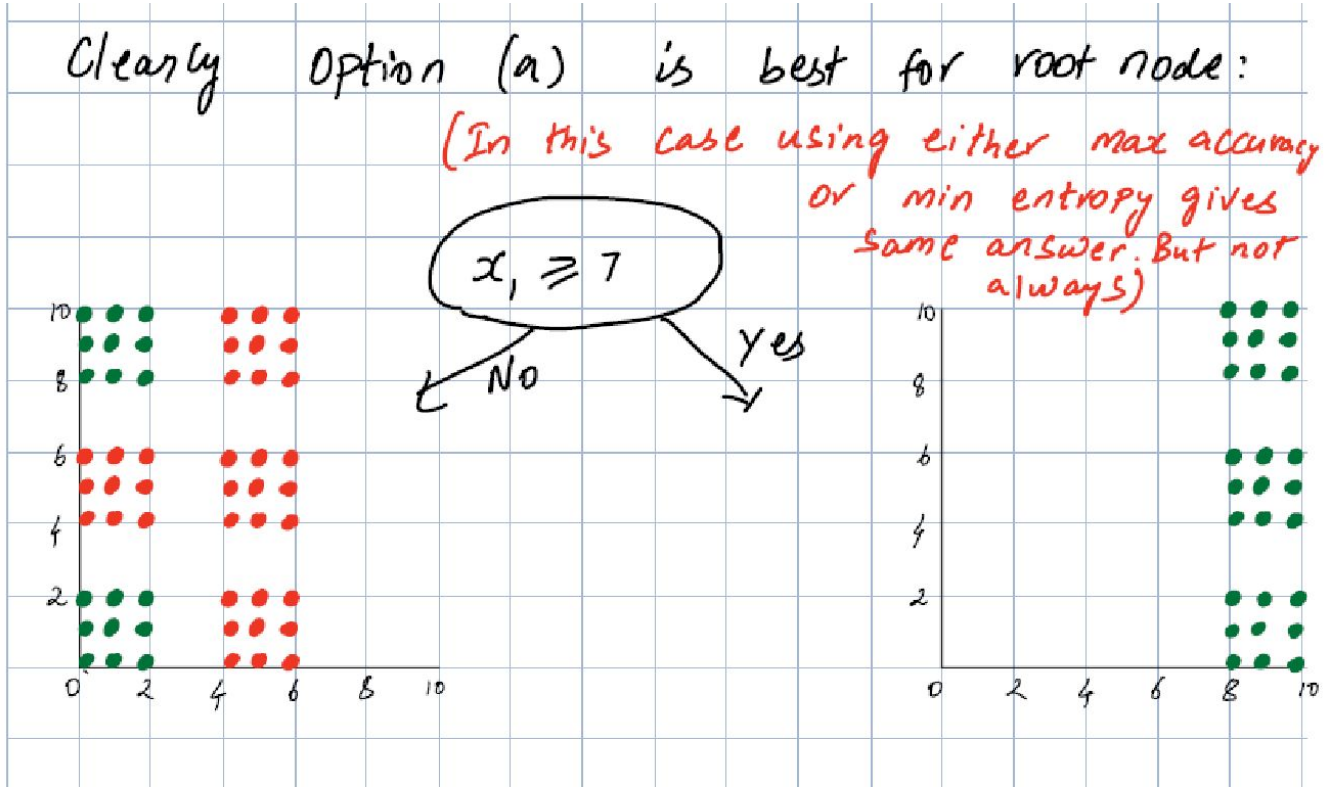
Illy accuracy of (b') which is the negation of above $= 1 - \frac{4}{9} = \frac{5}{9}$

L: $x_1 < 3$
R: $x_1 \geq 3$

Entropy of split b : $P_L H_L + P_R H_R$

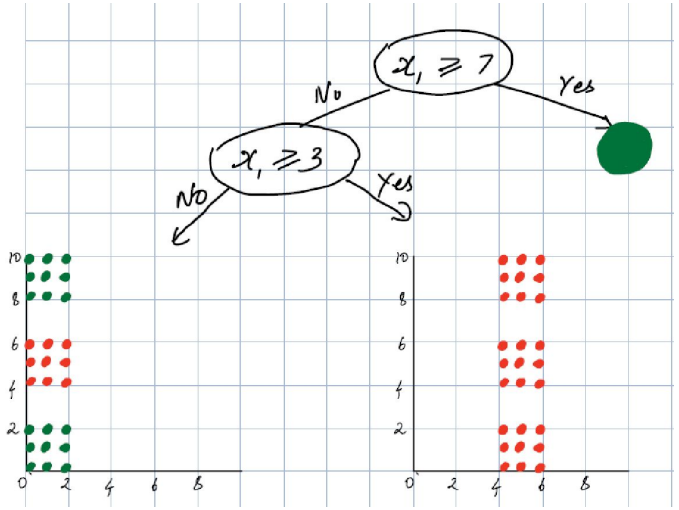
$P_L = \frac{1}{3}$ $q_L = \frac{2}{3}$: Entropy : $\frac{1}{3} H\left(\frac{2}{3}\right) + \frac{2}{3} H\left(\frac{1}{2}\right)$

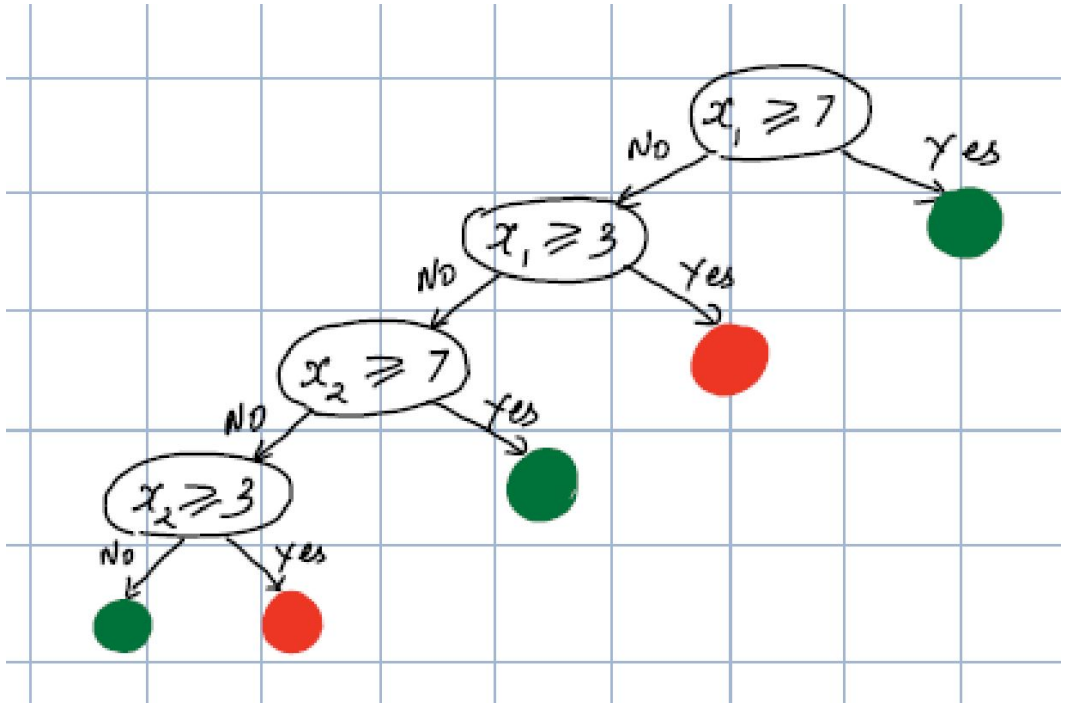
$P_R = \frac{2}{3}$ $q_R = \frac{1}{2}$



[HR]

Exercise: Recursively solve subproblems on left and right to derive this final tree



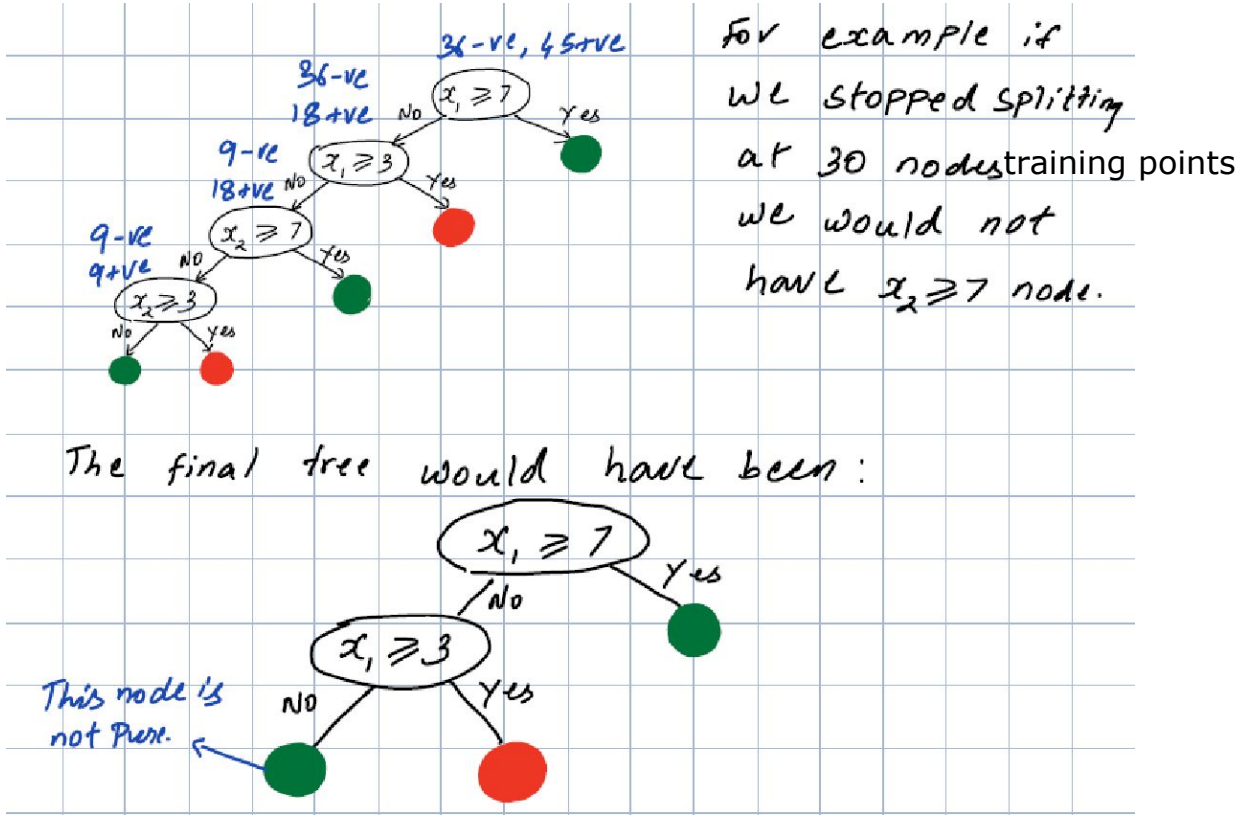


[HR]

Regularization of a decision tree

- Goal: Bring down the # of nodes in the decision tree without losing too much on accuracy.

- Solution: Stop early…
  - i) …at a certain depth of the tree, or
  - ii) …when number of training points is less than some number.



[HR]

Outline for Module M10

- M10. Combined models and Ensemble methods
  - M10.0 Introduction/Motivation
  - M10.1 Combined models
    - Conditional mixture models
    - Decision trees
  - **M10.3 Ensemble methods or Committee models**
    - Parallel ensemble methods (bagging)
    - Sequential ensemble methods (boosting)
  - M10.4 Parting thoughts

Committees

- Committees are ensemble methods that average the predictions of many individual learners

- Two very different approaches:
  - Bagging – average of **parallel**y/separately-trained **high-capacity** learners
  - Boosting – average of **sequential**ly/adaptively-trained **weak** learners

- Bias-variance decomposition helps in understanding certain aspects of bagging, and computational/statistical learning theory helps derive certain performance bounds on boosting.

[CMB]

Bias-variance analysis

$$h(\mathbf{x}) = \mathbb{E}[t|\mathbf{x}] = \int t p(t|\mathbf{x})\,\mathrm{d}t.$$

$$\mathbb{E}[L] = \int \{y(\mathbf{x}) - h(\mathbf{x})\}^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x} + \int \{h(\mathbf{x}) - t\}^2 p(\mathbf{x}, t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t.$$

$$\{y(\mathbf{x}; \mathcal{D}) - h(\mathbf{x})\}^2.$$

$$\begin{aligned}
&\{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] + \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\}^2 \\
&\quad = \{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})]\}^2 + \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\}^2 \\
&\qquad +2\{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})]\}\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\}.
\end{aligned}$$

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x}; \mathcal{D}) - h(\mathbf{x})\}^2\right] \\
&\quad = \underbrace{\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\}^2}_{(\text{bias})^2} + \underbrace{\mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})]\}^2\right]}_{\text{variance}}.
\end{aligned}$$

[CMB]

Bias-variance analysis summary

$$\text{expected loss} = (\text{bias})^2 + \text{variance} + \text{noise}$$

where

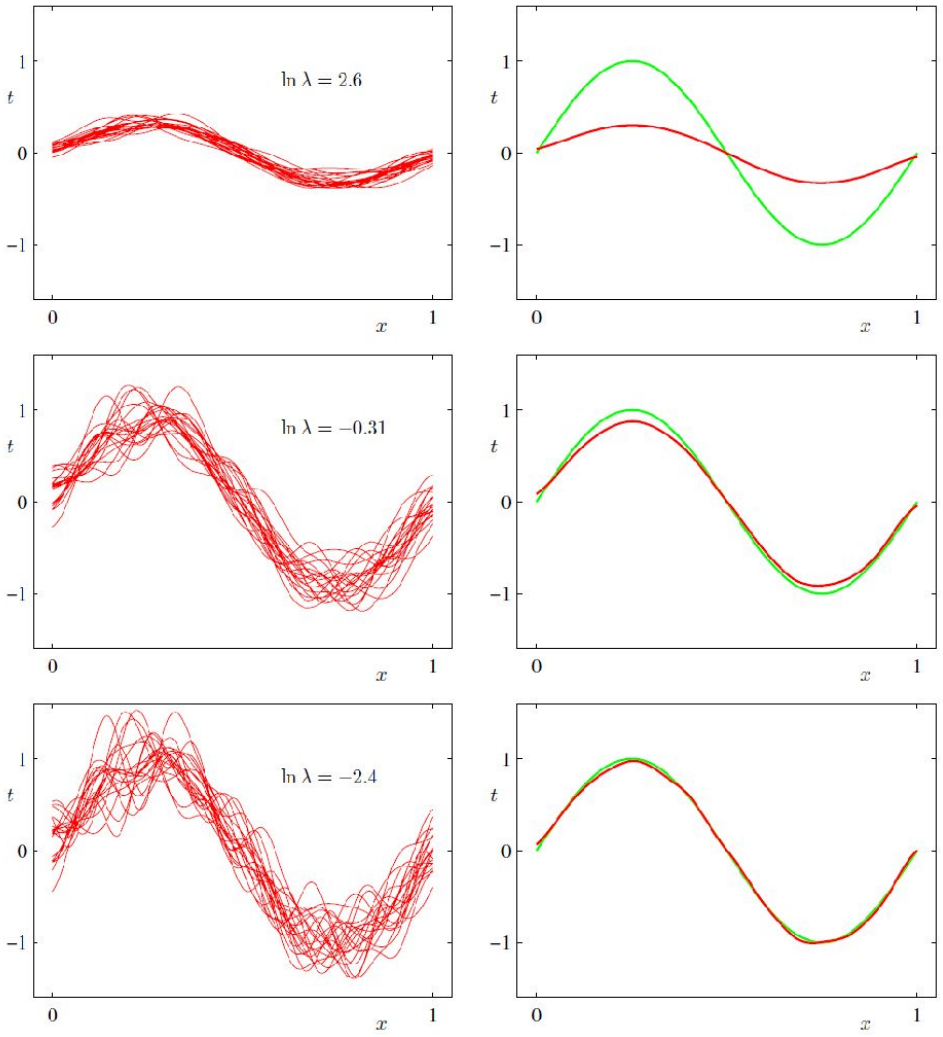
$$(\text{bias})^2 = \int \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x}$$

$$\text{variance} = \int \mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2\right] p(\mathbf{x})\,\mathrm{d}\mathbf{x}$$

$$\text{noise} = \int \{h(\mathbf{x}) - t\}^2 p(\mathbf{x}, t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t$$

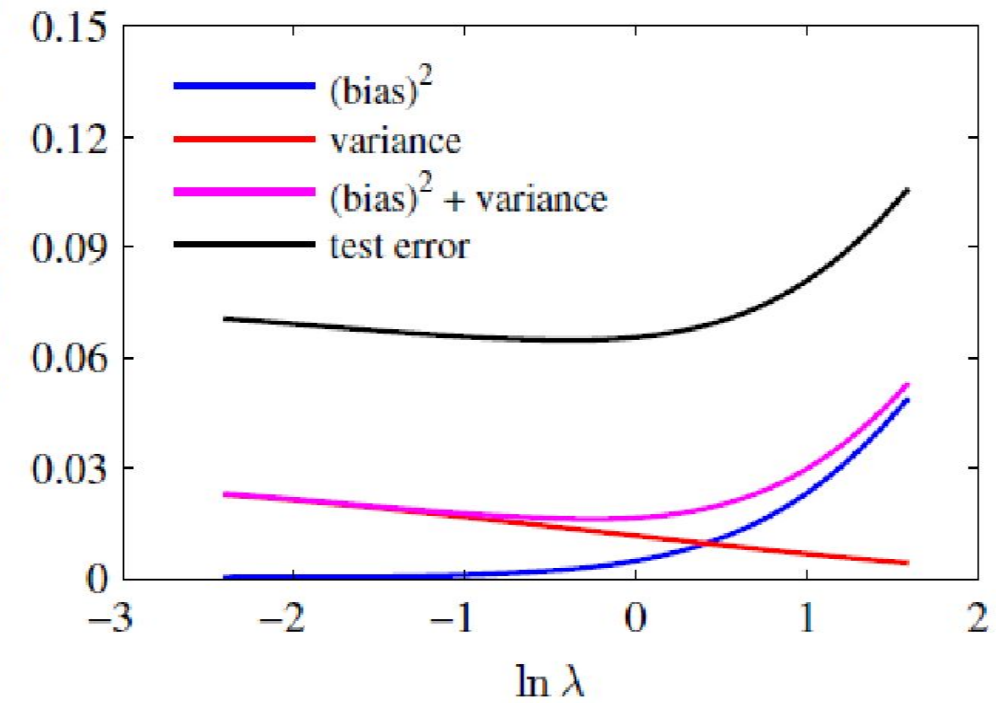
[CMB]

Bias-variance in pictures



[CMB]

Bias-variance analysis: empirical view



[CMB]

Bias-variance analysis: applicability in practice?

$$\text{expected loss} = (\text{bias})^2 + \text{variance} + \text{noise}$$

where

$$\begin{aligned}
(\text{bias})^2 &= \int \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x} \\
\text{variance} &= \int \mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2\right] p(\mathbf{x})\,\mathrm{d}\mathbf{x} \\
\text{noise} &= \int \{h(\mathbf{x}) - t\}^2 p(\mathbf{x},t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t
\end{aligned}$$

$$\overline{y}(x) = \frac{1}{L}\sum_{l=1}^{L} y^{(l)}(x)$$

$$\begin{aligned}
(\text{bias})^2 &= \frac{1}{N}\sum_{n=1}^{N} \{\overline{y}(x_n) - h(x_n)\}^2 \\
\text{variance} &= \frac{1}{N}\sum_{n=1}^{N} \frac{1}{L}\sum_{l=1}^{L} \{y^{(l)}(x_n) - \overline{y}(x_n)\}^2
\end{aligned}$$

[CMB]

Recall: Bias-variance analysis summary

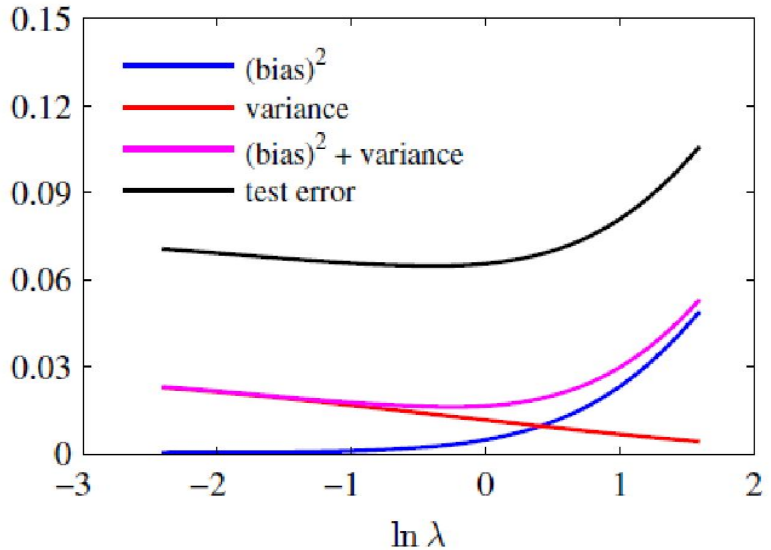
$$\text{expected loss} = (\text{bias})^2 + \text{variance} + \text{noise}$$

where

$$(\text{bias})^2 = \int \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x}$$

$$\text{variance} = \int \mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2\right] p(\mathbf{x})\,\mathrm{d}\mathbf{x}$$

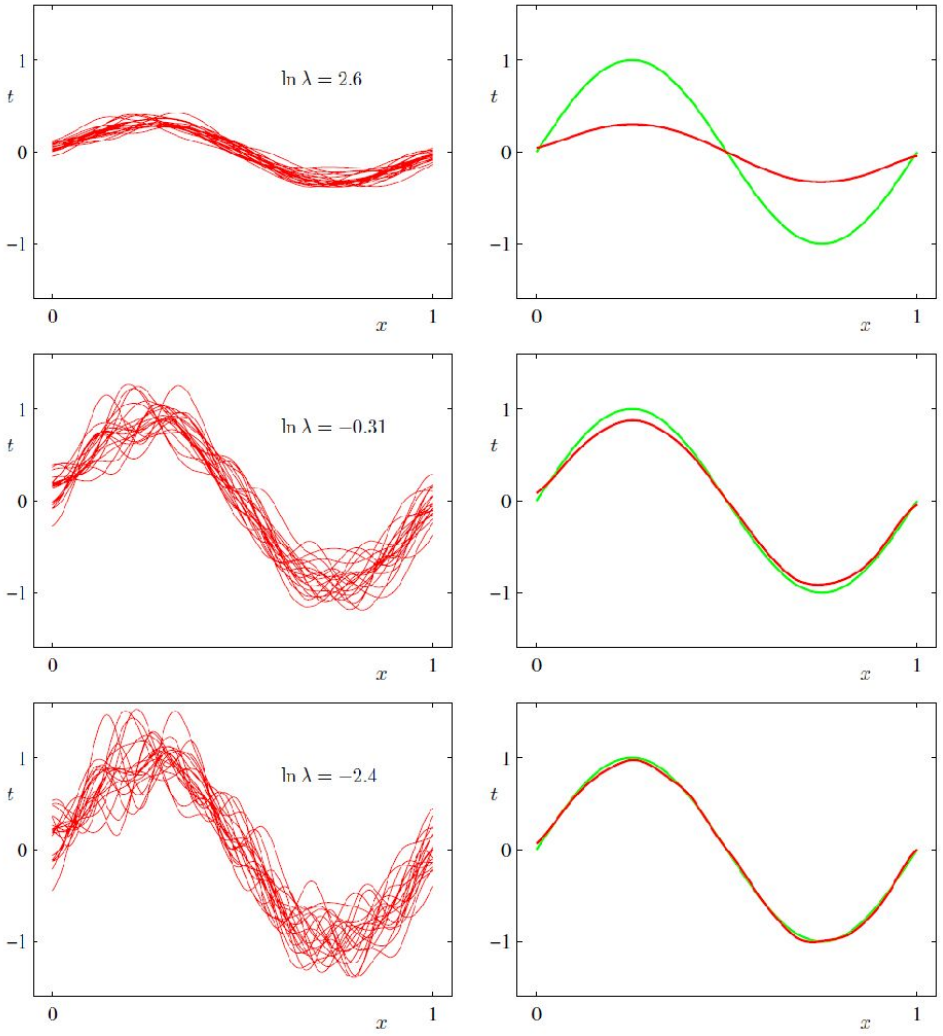
$$\text{noise} = \int \{h(\mathbf{x}) - t\}^2 p(\mathbf{x}, t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t$$



Exercise: cf. slides below or zoom quiz for careful understanding of what random variables the expectation above is taken over!

[CMB]

Recall: Bias-variance in pictures



[CMB]

Switch to Harish’s slides on Ensemble methods

[https://drive.google.com/file/d/1KUy2mziZ1pz-A0NeqlAV_qTP5Rq47Kkt/view?usp=sharing]

Outline for Module M10

- M10. Combined models and Ensemble methods
  - M10.0 Introduction/Motivation
  - M10.1 Combined models
    - Conditional mixture models
    - Decision trees
  - M10.3 Ensemble methods
    - Parallel ensemble methods (bagging)
    - Sequential ensemble methods (boosting)
  - **M10.4 Parting thoughts**

Recall: There and back again: Planned syllabus (tentative)

Subset of topics below to be covered (not necessarily in the same order):

1. Overview of PR/ML problems/algorithms (PR tasks/systems, ML paradigms)
2. Bayesian decision theory (Bayes classifier, loss functions)
3. Density estimation (Maximum likelihood, Bayesian estimation, Expectation Maximization (EM) for mixture density estimation, Non-parametric methods)
4. Linear models for classification and regression (Linear discriminant analysis (hyperplanes), Linear/polynomial regression, Bayesian regression)
5. Non-linear models for classification and regression (Support Vector Machines and kernel methods, Neural networks)
6. Combining models (Ensemble methods like boosting and bagging, Tree-based models)
7. Unsupervised learning methods for clustering and dimensionality reduction (E.g., hard/soft k-means clustering, Principal Component Analysis (PCA))
8. Select advanced topics based on time available (E.g., a subset of computational learning theory, algorithms for sequential data (Hidden Markov Models HMM), or graph-structured data; probabilistic graphical models).

Recall: A big question is:

WHY???

learn so many models/methods, if only one of the methods is all you hear about everywhere?

On a different context, always ask WHY you are taking this course? I hope it is not only campus placement opportunities or hype around ML, but also a general interest in understanding how you can take this field forward with your own creativity and internal ethical compass.

Parting thoughts…

Fun starts when you can take what you've learned from this course and apply it to choose the right model or right method of combining models in a systematic rather than brute-force fashion!

- Unified view of different models helps towards above goal – fixed vs. selective (SVM) vs. adaptive (ANN) basis functions; different paradigms, etc.
- Understanding conceptual/mathematical foundations of different methods also helps towards above goal.

Achieving machine intelligence (PRML) is hard, but we've

- a constructive argument for many specific problems, as seen in this course, and
- an existence proof for the general problem!

Real fun is in taking this field forward with your own creativity and internal ethical compass!

Thank you…

…for your interest in the course, your attention and your feedbacks!