Tuesday, 10/20/2020

# EE 660

# MACHINE LEARNING FROM SIGNALS: FOUNDATIONS AND METHODS

Prof. B. Keith Jenkins

**Lecture 17**

**Lecture 17** | **EE 660** | **Oct 20, 2020**

## Announcements

- Homework 6 (project proposal) is due Friday (10/23)

## Reading

- Intro ABM: Murphy 16.1 (last 2 paragraphs)
- CART and Random Forest: 16.2 (except 16.2.6)

## Today's topics

- Data Snooping
- Sampling Bias
- Adaptive Basis-Function Models (ABM)
- Classification and Regression Trees (CART) (part 1)

# Data Snooping [AML 5.3]

Basic principle: "If a data set has affected any step in the learning process, [then] it's ability to assess the outcome has been compromised."

Ex [AML Example 5.3]

Q: How much difference will snooping only for data normalization make?

Investment bank — goal: predict currency exchange rates
U.S. dollar ⟺ G.B. pound
$\mathcal{D}$: 8 years of historical data

Define ML problem: Predict direction of change for day $i$, given fluctuations in previous 20 days.

Procedure:

1. Standardize the entire dataset (to $\mu = 0$, $\sigma^2 = 1$)
2. Divide $\mathcal{D}$ → $\mathcal{D}_{Test}$ (set aside) (25%), → $\mathcal{D}'$ (for Training, validation) (75%)
3. Find best hypothesis $h_g$ using $\mathcal{D}'$
4. Evaluate $h_g$ using $\mathcal{D}_{Test}$

→ On $\mathcal{D}_{Test}$, it does well: 52.1 % correct.

⇒ Over 2 years of use, will give + 22% return on investment.

→ In reality, performed poorly (lost money)

Why? Conjecture: because $\mathcal{D}_{Test}$ was used to calculate normalizing parameters.

Verification: re-train the system using only training data to calculate normalizing parameters. $\mathcal{D}_{Test}$ is also normalized, using the parameters from $\mathcal{D}'$.

→ Performance on $\mathcal{D}_{Test}$ shows the system loses money.

## Two ways to deal with data snooping

1. Prevent it. Set $\mathcal{D}_{\text{Test}}$ aside at beginning; only use it at end (after training and decisions / choices have been made).

2. Account for it. Use our bounds on $E_{out}$, in terms of $\varepsilon_{eff}$, $\varepsilon_{VC}$, or $\varepsilon_M$, as a guide on the amount of "contamination". With some care (e.g., small $M$), it can be kept minimal.

## Sampling Bias [AML 5.2]

Ex: 1948 U.S. presidential election. Mr. Truman vs. Mr. Dewey.

Telephone poll on night of election ("Who did you vote for?")
$\rightarrow$ Dewey was ahead by more than their error bar.
$\Rightarrow$ Major newspaper ran an article claiming Dewey won.

After votes were counted $\Rightarrow$ Truman won.

What went wrong?

→ Only high-income households had telephones

∴ $\mathcal{D}$ came from $p_{\mathcal{D}}(\underline{x}')$, which sampled mostly high-income households.

$p(\underline{x}')$ was the pdf of all voters, and $p_{\mathcal{D}}(\underline{x}') \neq p(\underline{x}')$.

[ Sampling bias occurs when the pdf the dataset is drawn from, $p_{\mathcal{D}}(\underline{x}')$, differs from the true pdf of the problem (or unknowns), $p(\underline{x}')$.

Comments:

1. If $p_{\mathcal{D}}(\underline{x}')$ and $p(\underline{x}')$ are known or can be estimated, then there are ways to compensate for sampling bias (N.R.F.)
   - → In ML, "domain adaptation"
   - → In statistics, very common.

2. In the above, $\underline{x}'$ refers to all variables that can affect the outcome, including any relevant variables that are not in the input feature set $\underline{x}$.

## Adaptive Basis-function Models (ABM) [Murphy 16.1, last 2 paragr.'s]

$$\hat{f}(\underline{x}) = w_0 + \sum_{m=1}^{M} w_m \phi_m(\underline{x}) \qquad (\underline{w} = \underline{w}^{(0)}) \qquad (16.3)$$

in which $\phi_m(\underline{x})$ is learned from the data

If the $\phi_m(\underline{x})$ are parametric, then:

$$\phi_m(\underline{x}) = \phi(\underline{x}; \underline{v}_m)$$

{ parameters of $\phi_m$, to be learned from the data.

## Classification and Regression Trees (CART) [Murphy 16.2]

(also called "decision trees")

Model: $$\hat{f}(\underline{x}) = \sum_{m=1}^{M} w_m \mathbb{I}(\underline{x} \in R_m) = \sum_{m=1}^{M} w_m \phi(\underline{x}; \underline{v}_m)$$

Regression case: $w_m$ = value of $\hat{f}$ in $R_m$

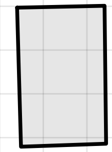
indicator fcn. $\mathbb{I}[\cdot]$

$m^{\text{th}}$ region

$\Rightarrow$ $\hat{f}(\underline{x})$ is a piecewise-constant fcn. of $\underline{x}$. (approx. to $f(\underline{x})$).

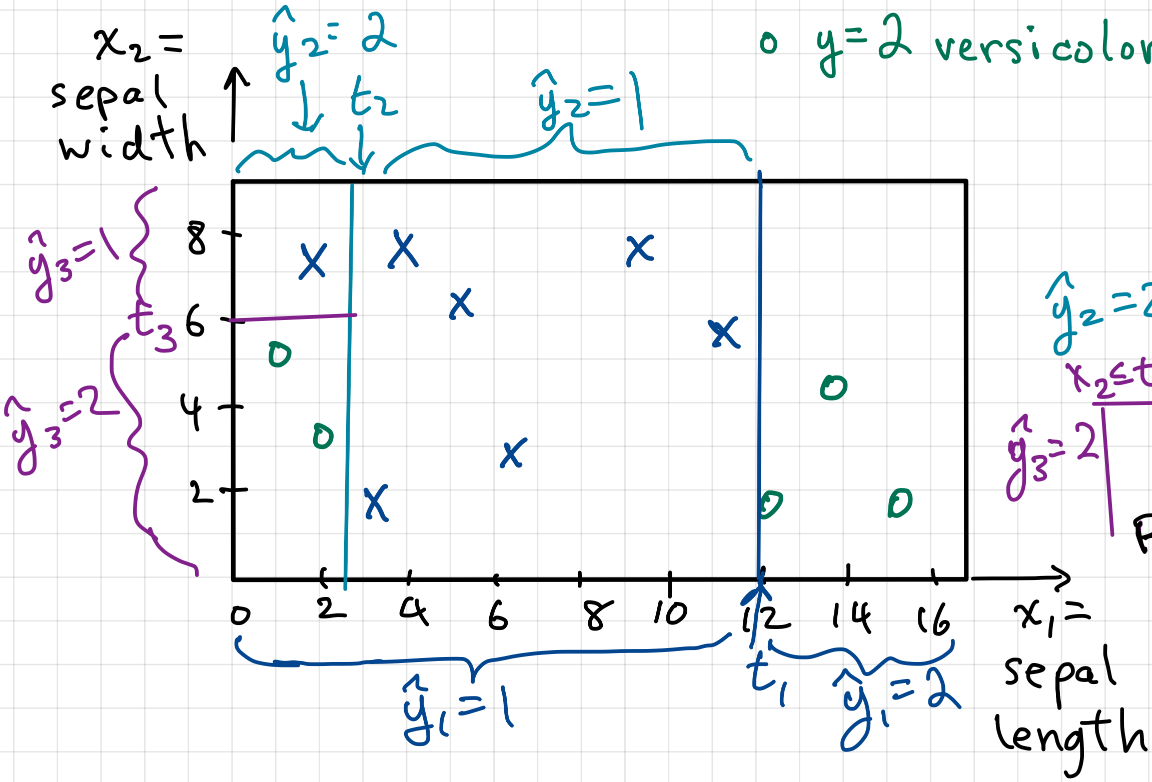
CART forms a tree, and a set of regions $R_m$ in feature space.

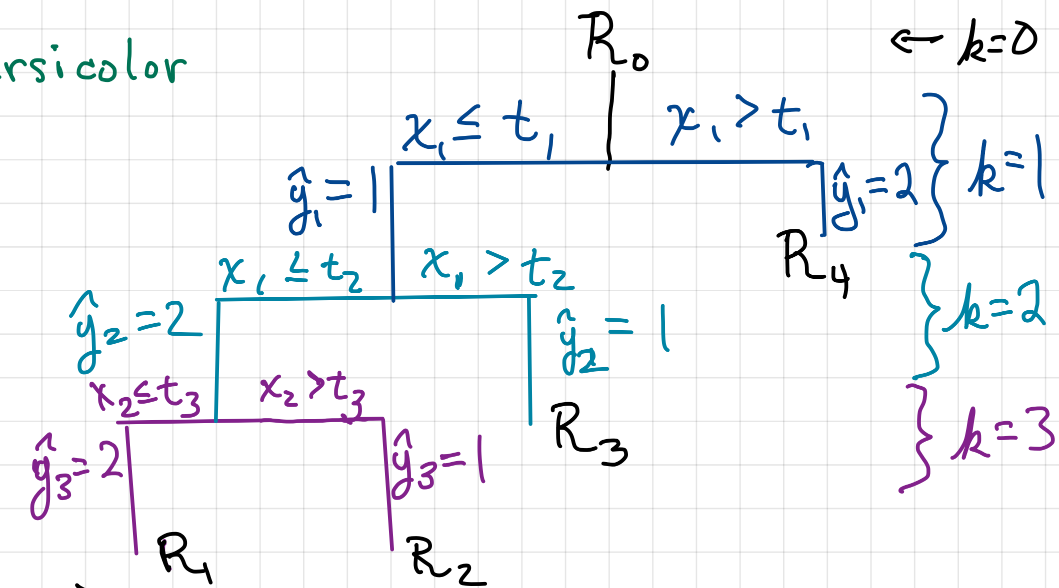
# CART Example: classification

$\phi_m$ : 

× $y=1$ setosa

o $y=2$ versicolor




Error measure: % misclassified points.

Minimize $E_{in}$ at each iteration.

Start: $k=0$

$R_0$: all feature space $\in \hat{y} = 1$

$E_{in} = 5/12$.

| Iteration $k$ | $E_{in}(k)$ |
|---|---|
| 0 | 5/12 |
| 1 | 2/12 |
| 2 | 1/12 |
| 3 | 0/12 |

At each iteration:

Choose:

- Region $R_m$ to split
- one feature ($x_1$ or $x_2$) to threshold
- thereshold value $t_k$
- region labels

Calculate error $E_{in}(k)$.

There exist a variety of halting conditions, such as:

- Max. depth of tree
- Min. reduction of cost (error) fcn. to split a region.
- Min. # of data pts. in a final region.

[ref: Murphy]