CS 171: Intro to ML and DM

Christian Shelton

UC Riverside

Slide Set 11: Decision Trees I

Slides from CS 171

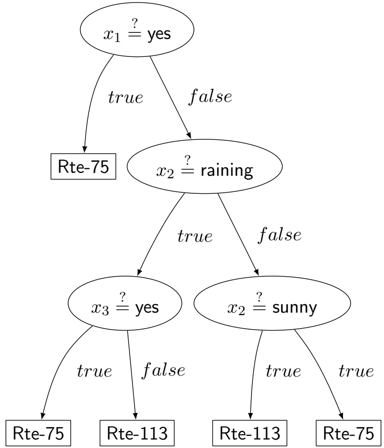
- From UC Riverside
  - CS 171: Introduction to Machine Learning and Data Mining
  - Professor Christian Shelton
- DO NOT REDISTRIBUTE
  - These slides contain copyrighted material (used with permission) from
    - Elements of Statistical Learning (Hastie, et al.)
    - Pattern Recognition and Machine Learning (Bishop)
    - An Introduction to Machine Learning (Kubat)
    - Machine Learning: A Probabilistic Perspective (Murphy)
  - For use only by enrolled students in the course

A toy problem

x_1(weekend?)	x_2(weather)	x_3(game?)	y(faster route)
no	sunny	no	Rte-113
no	sunny	yes	Rte-113
no	cloudy	yes	Rte-75
yes	sunny	no	Rte-113
no	raining	no	Rte-113
no	raining	yes	Rte-75
yes	cloudy	yes	Rte-75
no	sunny	no	Rte-113
yes	sunny	no	Rte-75
no	cloudy	no	Rte-113

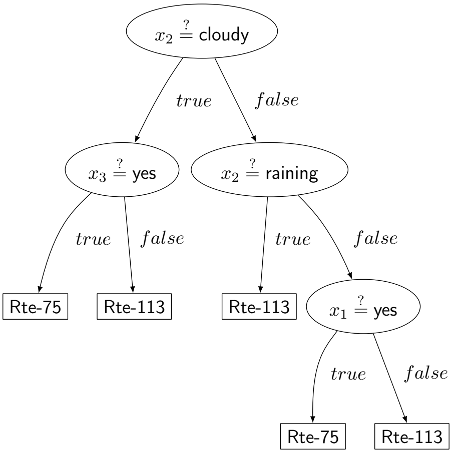
A toy decision tree

x_1(weekend?)	x_2(weather)	x_3(game?)	y(faster route)
no	sunny	no	Rte-113
no	sunny	yes	Rte-113
no	cloudy	yes	Rte-75
yes	sunny	no	Rte-113
no	raining	no	Rte-113
no	raining	yes	Rte-75
yes	cloudy	yes	Rte-75
no	sunny	no	Rte-113
yes	sunny	no	Rte-75
no	cloudy	no	Rte-113



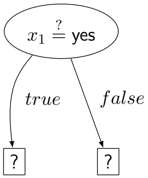
A toy decision tree

x_1(weekend?)	x_2(weather)	x_3(game?)	y(faster route)
no	sunny	no	Rte-113
no	sunny	yes	Rte-113
no	cloudy	yes	Rte-75
yes	sunny	no	Rte-113
no	raining	no	Rte-113
no	raining	yes	Rte-75
yes	cloudy	yes	Rte-75
no	sunny	no	Rte-113
yes	sunny	no	Rte-75
no	cloudy	no	Rte-113



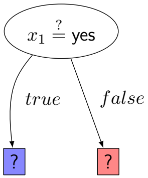
Greedy Decision Tree Learning

x_1(weekend?)	x_2(weather)	x_3(game?)	y(faster route)
no	sunny	no	Rte-113
no	sunny	yes	Rte-113
no	cloudy	yes	Rte-75
yes	sunny	no	Rte-113
no	raining	no	Rte-113
no	raining	yes	Rte-75
yes	cloudy	yes	Rte-75
no	sunny	no	Rte-113
yes	sunny	no	Rte-75
no	cloudy	no	Rte-113



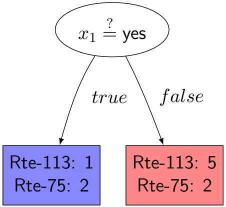
Greedy Decision Tree Learning

x_1(weekend?)	x_2(weather)	x_3(game?)	y(faster route)
no	sunny	no	Rte-113
no	sunny	yes	Rte-113
no	cloudy	yes	Rte-75
yes	sunny	no	Rte-113
no	raining	no	Rte-113
no	raining	yes	Rte-75
yes	cloudy	yes	Rte-75
no	sunny	no	Rte-113
yes	sunny	no	Rte-75
no	cloudy	no	Rte-113



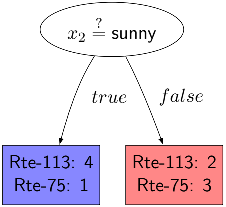
Greedy Decision Tree Learning

x_1(weekend?)	x_2(weather)	x_3(game?)	y(faster route)
no	sunny	no	Rte-113
no	sunny	yes	Rte-113
no	cloudy	yes	Rte-75
yes	sunny	no	Rte-113
no	raining	no	Rte-113
no	raining	yes	Rte-75
yes	cloudy	yes	Rte-75
no	sunny	no	Rte-113
yes	sunny	no	Rte-75
no	cloudy	no	Rte-113



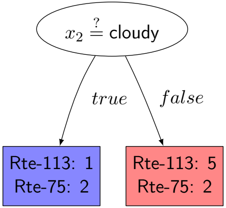
Greedy Decision Tree Learning

x_1(weekend?)	x_2(weather)	x_3(game?)	y(faster route)
no	sunny	no	Rte-113
no	sunny	yes	Rte-113
no	cloudy	yes	Rte-75
yes	sunny	no	Rte-113
no	raining	no	Rte-113
no	raining	yes	Rte-75
yes	cloudy	yes	Rte-75
no	sunny	no	Rte-113
yes	sunny	no	Rte-75
no	cloudy	no	Rte-113



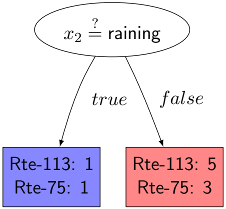
Greedy Decision Tree Learning

x_1(weekend?)	x_2(weather)	x_3(game?)	y(faster route)
no	sunny	no	Rte-113
no	sunny	yes	Rte-113
no	cloudy	yes	Rte-75
yes	sunny	no	Rte-113
no	raining	no	Rte-113
no	raining	yes	Rte-75
yes	cloudy	yes	Rte-75
no	sunny	no	Rte-113
yes	sunny	no	Rte-75
no	cloudy	no	Rte-113



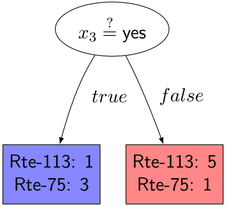
Greedy Decision Tree Learning

x_1(weekend?)	x_2(weather)	x_3(game?)	y(faster route)
no	sunny	no	Rte-113
no	sunny	yes	Rte-113
no	cloudy	yes	Rte-75
yes	sunny	no	Rte-113
no	raining	no	Rte-113
no	raining	yes	Rte-75
yes	cloudy	yes	Rte-75
no	sunny	no	Rte-113
yes	sunny	no	Rte-75
no	cloudy	no	Rte-113



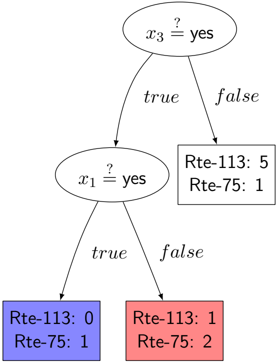
Greedy Decision Tree Learning

x_1(weekend?)	x_2(weather)	x_3(game?)	y(faster route)
no	sunny	no	Rte-113
no	sunny	yes	Rte-113
no	cloudy	yes	Rte-75
yes	sunny	no	Rte-113
no	raining	no	Rte-113
no	raining	yes	Rte-75
yes	cloudy	yes	Rte-75
no	sunny	no	Rte-113
yes	sunny	no	Rte-75
no	cloudy	no	Rte-113



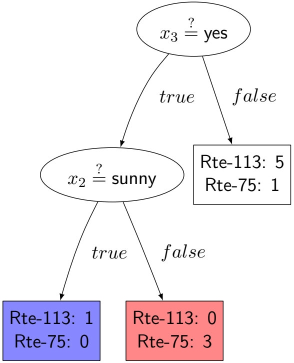
Greedy Decision Tree Learning

x_1(weekend?)	x_2(weather)	x_3(game?)	y(faster route)
no	sunny	no	Rte-113
no	sunny	yes	Rte-113
no	cloudy	yes	Rte-75
yes	sunny	no	Rte-113
no	raining	no	Rte-113
no	raining	yes	Rte-75
yes	cloudy	yes	Rte-75
no	sunny	no	Rte-113
yes	sunny	no	Rte-75
no	cloudy	no	Rte-113



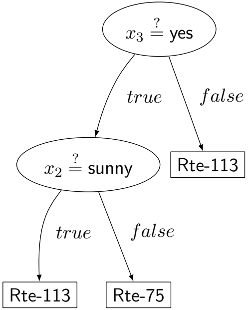
Greedy Decision Tree Learning

x_1(weekend?)	x_2(weather)	x_3(game?)	y(faster route)
no	sunny	no	Rte-113
no	sunny	yes	Rte-113
no	cloudy	yes	Rte-75
yes	sunny	no	Rte-113
no	raining	no	Rte-113
no	raining	yes	Rte-75
yes	cloudy	yes	Rte-75
no	sunny	no	Rte-113
yes	sunny	no	Rte-75
no	cloudy	no	Rte-113



Greedy Decision Tree Learning

x_1(weekend?)	x_2(weather)	x_3(game?)	y(faster route)
no	sunny	no	Rte-113
no	sunny	yes	Rte-113
no	cloudy	yes	Rte-75
yes	sunny	no	Rte-113
no	raining	no	Rte-113
no	raining	yes	Rte-75
yes	cloudy	yes	Rte-75
no	sunny	no	Rte-113
yes	sunny	no	Rte-75
no	cloudy	no	Rte-113



Decision Tree Learning Overview

Given a data set X, and Y

1. If no test possible, or all Ys are the same,
   Return tree of a single leaf (the majority class in Y).
2. Otherwise,
   1. Select the binary test (of x) that best separates the ys
   2. Let X_t and Y_t be the examples for which the test is true.
   3. Let X_f and Y_f be the examples for which the test is false.
   4. Recursively call on (X_t, Y_t), assigning result to T_t.
   5. Recursively call on (X_f, Y_f), assigning result to T_f.
   6. Return tree of binary test, with T_t on true branch and T_f on false branch.

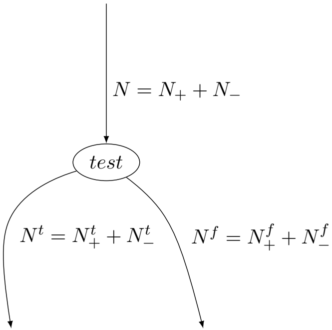
Decision Tree Learning Overview

Given a data set X, and Y

1. If no test possible, or all Ys are the same,
   Return tree of a single leaf (the majority class in Y).
2. Otherwise,
   1. Select the binary test (of x) that best separates the ys
   2. Let X_t and Y_t be the examples for which the test is true.
   3. Let X_f and Y_f be the examples for which the test is false.
   4. Recursively call on (X_t, Y_t), assigning result to T_t.
   5. Recursively call on (X_f, Y_f), assigning result to T_f.
   6. Return tree of binary test, with T_t on true branch and T_f on false branch.

So what is "best?"

Scoring a Decision Node



Tempting to use number of errors:

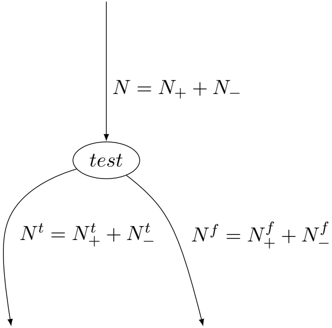
$$\begin{aligned}
&\min(N^t_-, N^t_+) + \min(N^f_-, N^f_+) \\
&\quad = N^t \min\left(\frac{N^t_-}{N^t}, \frac{N^t_+}{N^t}\right) + N^f \min\left(\frac{N^f_-}{N^f}, \frac{N^f_+}{N^f}\right) \\
&\quad = N^t \mathsf{score}_{\mathsf{error}}(p^t_-, p^t_+) + N^f \mathsf{score}_{\mathsf{error}}(p^f_-, p^f_+)
\end{aligned}$$

where

$$p^t_- = \frac{N^t_-}{N^t} \qquad p^t_+ = \frac{N^t_+}{N^t}$$

$$p^f_- = \frac{N^f_-}{N^f} \qquad p^f_+ = \frac{N^f_+}{N^f}$$

Scoring a Decision Node



Tempting to use number of errors:

$$\begin{aligned}
\min(N^t_-, N^t_+) + \min(N^f_-, N^f_+) \\
&= N^t \min\left(\frac{N^t_-}{N^t}, \frac{N^t_+}{N^t}\right) + N^f \min\left(\frac{N^f_-}{N^f}, \frac{N^f_+}{N^f}\right) \\
&= N^t \text{score}_{\text{error}}(p^t_-, p^t_+) + N^f \text{score}_{\text{error}}(p^f_-, p^f_+)
\end{aligned}$$

where

$$p^t_- = \frac{N^t_-}{N^t} \qquad p^t_+ = \frac{N^t_+}{N^t}$$

$$p^f_- = \frac{N^f_-}{N^f} \qquad p^f_+ = \frac{N^f_+}{N^f}$$

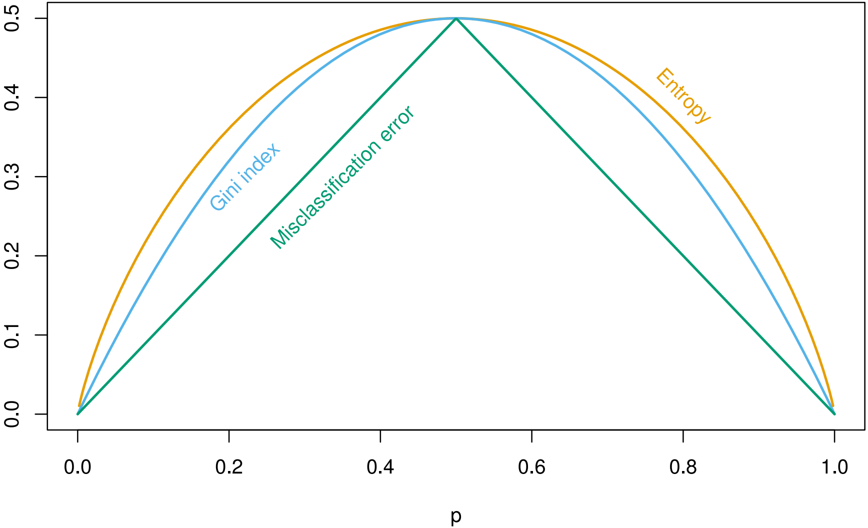
But this does not account for the later refinement to each branch.

Scoring a Decision Node

Possible scores:

- $\text{score}_{\text{error}}(p_-, p_+) = \min(p_-, p_+)$ (misclassification rate)
- $\text{score}_{\text{Gini}}(p_-, p_+) = p_- p_+$ (Gini index)
- $\text{score}_{\text{entropy}}(p_-, p_+) = -p_- \ln p_- - p_+ \ln p_+$ (Cross-entropy)

Scoring a Decision Node



Continuous Features

What about features that are not categorical?

Continuous Features

What about features that are not categorical?

If x_1 is a real-valued feature, we consider tests of the form

$$x_1 < t$$

for different values of t.

Continuous Features

What about features that are not categorical?

If x_1 is a real-valued feature, we consider tests of the form

$$x_1 < t$$

for different values of t.

feature type	number values	number tests
categorical	k	k
continuous	∞	???

Continuous Features

What about features that are not categorical?

If x_1 is a real-valued feature, we consider tests of the form

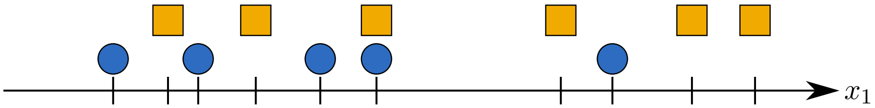
$$x_1 < t$$

for different values of t.

feature type	number values	number tests
categorical	k	k
continuous	∞	$\leq m$

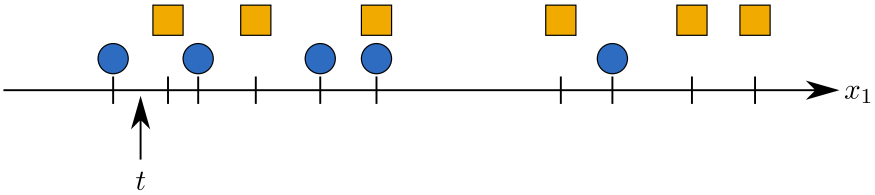
Continuous Features

Consider splitting the dataset on feature x_1 (for instance).
Plotting only x_1 versus y:



Continuous Features

Consider splitting the dataset on feature x_1 (for instance).
Plotting only x_1 versus y:

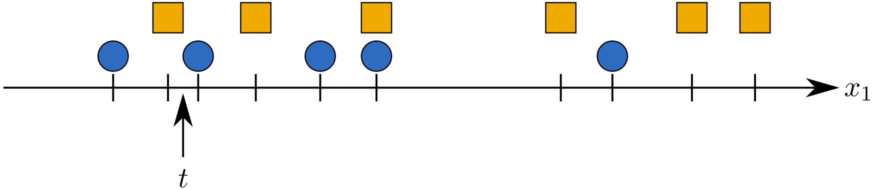


N	$<$	$>$
+ ■	0	6
− ●	1	4

p	$<$	$>$
+ ■	0/1	6/10
− ●	1/1	4/10

Continuous Features

Consider splitting the dataset on feature x_1 (for instance).
Plotting only x_1 versus y:

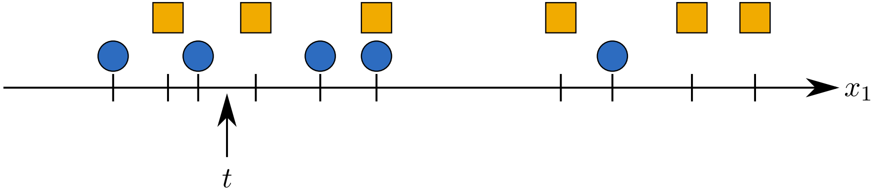


N	$<$	$>$
$+$	1	5
$-$	1	4

p	$<$	$>$
$+$	1/2	5/9
$-$	1/2	4/9

Continuous Features

Consider splitting the dataset on feature x_1 (for instance).
Plotting only x_1 versus y:

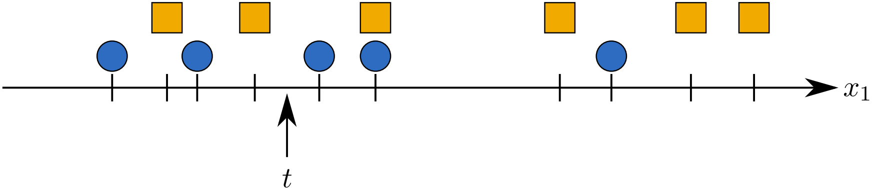


N	<	>
+ ■	1	5
− ●	2	3

p	<	>
+ ■	1/3	5/8
− ●	2/3	3/8

Continuous Features

Consider splitting the dataset on feature x_1 (for instance).
Plotting only x_1 versus y:

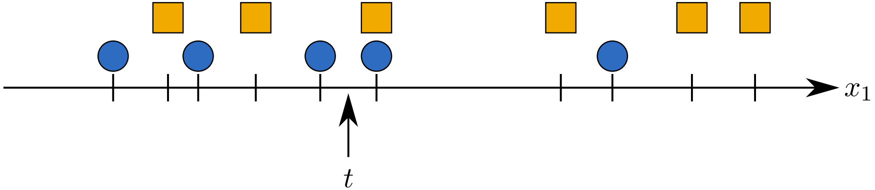


N	$<$	$>$
$+$	2	4
$-$	2	3

p	$<$	$>$
$+$	2/4	4/7
$-$	2/4	3/7

Continuous Features

Consider splitting the dataset on feature x_1 (for instance).
Plotting only x_1 versus y:

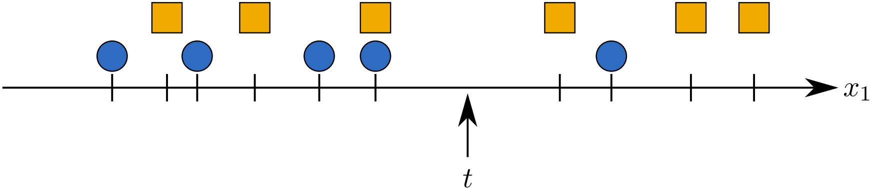


N	$<$	$>$
+ ■	2	4
− ●	3	2

p	$<$	$>$
+ ■	2/5	4/6
− ●	3/5	2/6

Continuous Features

Consider splitting the dataset on feature x_1 (for instance).
Plotting only x_1 versus y:

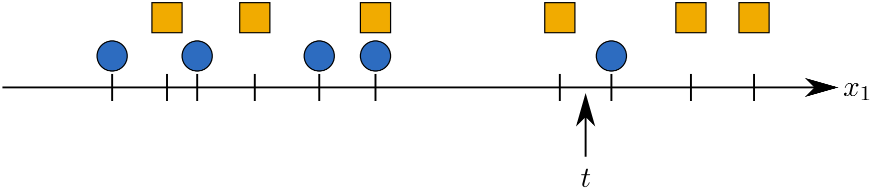


N	<	>
+ ■	3	3
− ●	4	1

p	<	>
+ ■	3/7	3/4
− ●	4/7	1/4

Continuous Features

Consider splitting the dataset on feature x_1 (for instance).
Plotting only x_1 versus y:

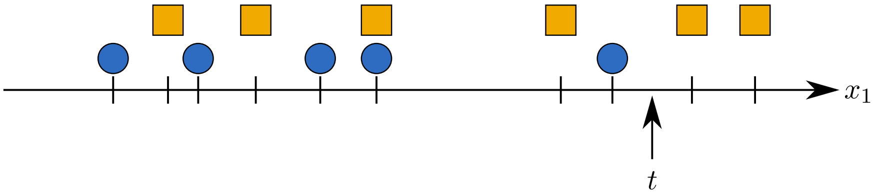


N	$<$	$>$
$+$ ■	4	2
$-$ ●	4	1

p	$<$	$>$
$+$ ■	4/8	2/3
$-$ ●	4/8	1/3

Continuous Features

Consider splitting the dataset on feature x_1 (for instance).
Plotting only x_1 versus y:

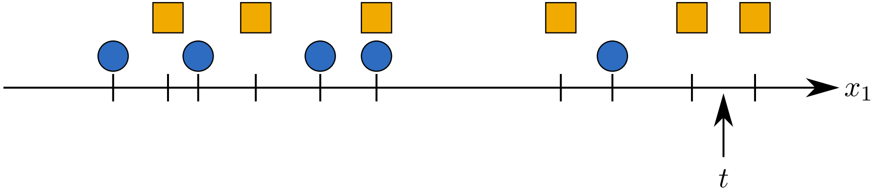


N	$<$	$>$
+ ■	4	2
− ●	5	0

p	$<$	$>$
+ ■	4/9	2/2
− ●	5/9	0/2

Continuous Features

Consider splitting the dataset on feature x_1 (for instance).
Plotting only x_1 versus y:

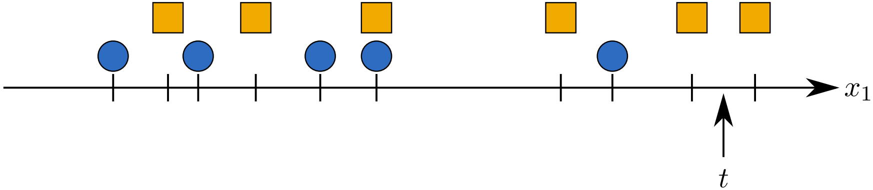


N	$<$	$>$
$+$ ■	5	1
$-$ ●	5	0

p	$<$	$>$
$+$ ■	5/10	1/1
$-$ ●	5/10	0/1

Continuous Features

Consider splitting the dataset on feature x_1 (for instance).
Plotting only x_1 versus y:



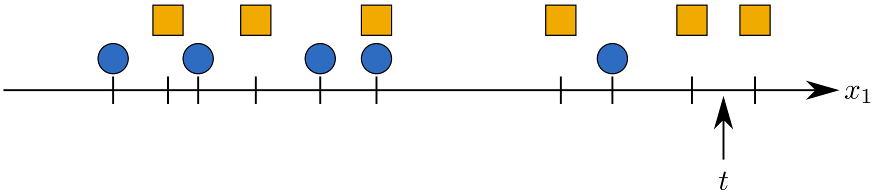
N	$<$	$>$
+ ■	5	1
− ●	5	0

p	$<$	$>$
+ ■	5/10	1/1
− ●	5/10	0/1

So, with n_c categorical features, each with k values and n_r real-valued features, how many tests must the algorithm check for each split?

Continuous Features

Consider splitting the dataset on feature x_1 (for instance).
Plotting only x_1 versus y:



N	$<$	$>$
$+$ ■	5	1
$-$ ●	5	0

p	$<$	$>$
$+$ ■	5/10	1/1
$-$ ●	5/10	0/1

So, with n_c categorical features, each with k values and n_r real-valued features, how many tests must the algorithm check for each split?
Answer: $n_c \times k + n_r \times m$