amazon | science

Tushaar Gangavarapu

ML Research Engineer
Kindle Content Experience, AQuA
Amazon.com, Inc.

MLU Keynote – Oct. 12, 2020

Decision Trees

Learning to Predict

amazon | science

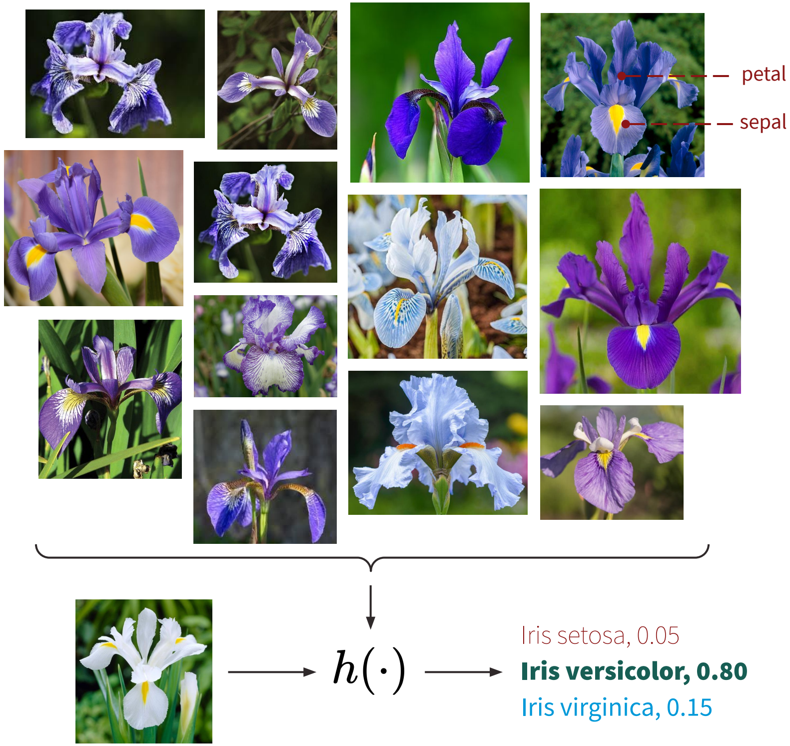
whoami

tusgan@amazon.com

Agenda

1. Moving From k-NN to Decision Trees
2. Region-based Loss Estimation
3. Extensions on Decision Trees
4. Specific Use Cases of Decision Trees
5. Concluding Remarks

Problem Setting: Supervised Learning



- Given a bunch of **training examples**, with each training example **annotated**:
  - **Learn** the patterns in known data
  - **Generalize** on unseen data
- **Hypothesis**: transformation from input features to output values
- **Classification** *vs*. **regression**: countably discrete *vs*. continuous outputs
- Bias-variance tradeoff – **bias = paying no attention to training data**; **variance = paying too much attention to training data**

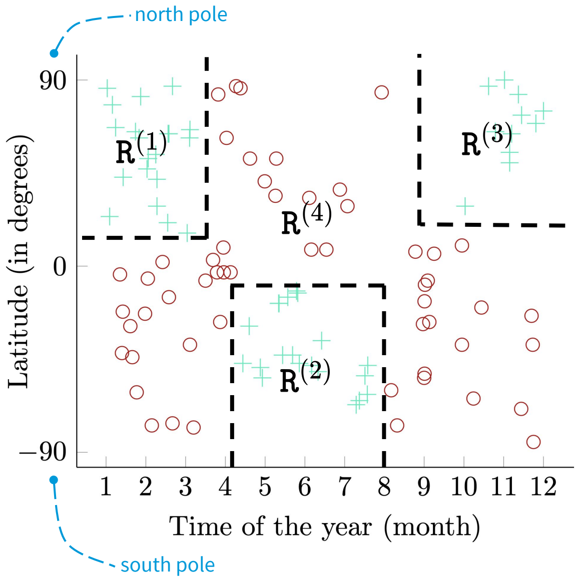
RA Fisher. *The use of multiple measurements in taxonomic problems.* Annual Eugenics, Part II. pages 179–188, 1936

Agenda

1. Moving From *k*-NN to Decision Trees
2. Region-based Loss Estimation
3. Extensions on Decision Trees
4. Specific Use Cases of Decision Trees
5. Concluding Remarks

Motivation: Accommodating Nonlinearity

- Do linear models: GLMs, *v*-SVMs handle **data nonlinearity**?
  - **Linear hypothesis**: $h(x) = h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \cdots + \theta_n x_n = \sum_{j=0}^{n} \theta_j x_j$
  - **Linear decision boundary**: logistic regression *vs*. neural networks?
  - Kernelization trick *vs*. **decision trees**: feature mapping
- Explore the **inherent structure** in the data: moving from *k*–NN to decision trees
  - ***k*–NN**: clusters of homogeneous class alignments
  - For a given input, $x^{(i)}$, if we somehow knew that $x^{(i)}$ belongs to a **cluster**?
  - **Relevance**: exact identity of $x^{(i)}$ *vs*. **cluster knowledge**
- **Decision trees**: determine (non-overlapping) areas of interest



Learning problem: predict whether to ski or not, given the location and time of year

$$\mathbf{X} = \bigcup_{j=1}^{r} R^{(j)}, \ r \in \mathbb{Z}^+$$

Raphael John Lamarre Townshend. *Decision trees*. pages 2–3, 2019

Choosing Regions: Top-Down, Recursive, Greedy

- How to **split** the input space into regions of interest?
  - **Occam's razor**: *non sunt multiplicanda entia praeter necessitatem* — max compact regions
  - **NP-hard**: max compact α exact cover by three sets
  - Decision trees: **top-down**, **recursive**, **greedy** partitions
- Playing *twenty questions* with the data: space partitioned into **homogeneous clusters** (w.r.t. class)[majority vote]

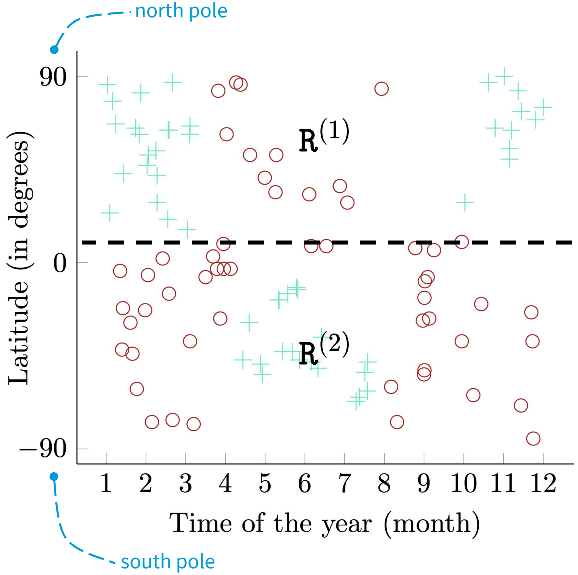
threshold on 's'

splitting attribute

$$\text{split}(s,t) = \left(\mathtt{R}^{(\text{parent}.1)} = \left\{x^{(i)} \mid x_s^{(i)} < t, x^{(i)} \in \mathtt{R}^{(\text{parent})}\right\},\ \mathtt{R}^{(\text{parent}.2)} = \left\{x^{(i)} \mid x_s^{(i)} \geq t, x^{(i)} \in \mathtt{R}^{(\text{parent})}\right\}\right)$$

- **Unseen data**: traverse from root to leaf; satisfying criteria on set at the internal nodes of the tree — $[O(\log_2 m); O(m)]$

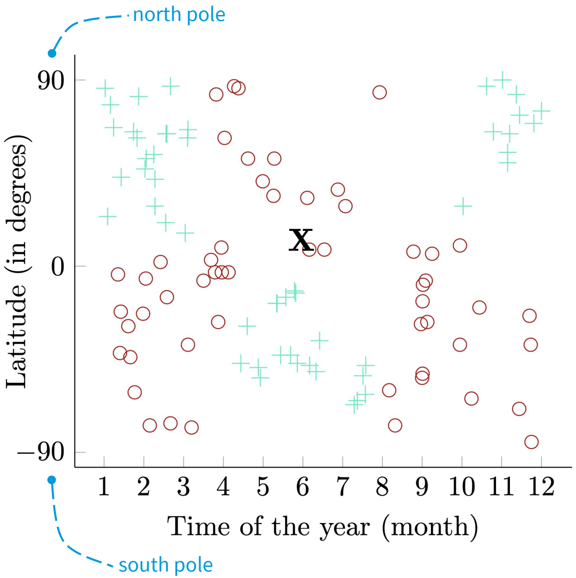
Anselm Blumer, et al. *Occam's razor*. Information processing letters, 24(6):377–380, 1987



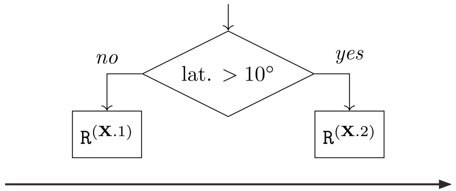
Learning problem: predict whether to ski or not, given the location and time of year

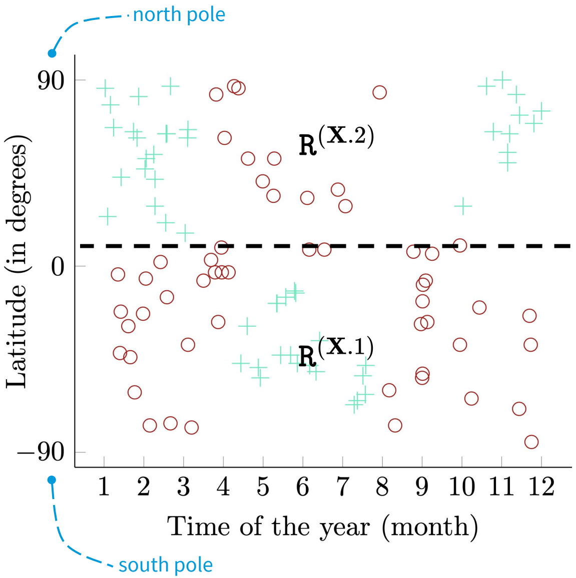
$$\mathbf{X} = \bigcup_{j=1}^{r} \mathtt{R}^{(j)},\ r \in \mathbb{Z}^+$$

Choosing Regions: Top-Down, Recursive, Greedy



Learning problem: predict whether to ski or not, given the location and time of year

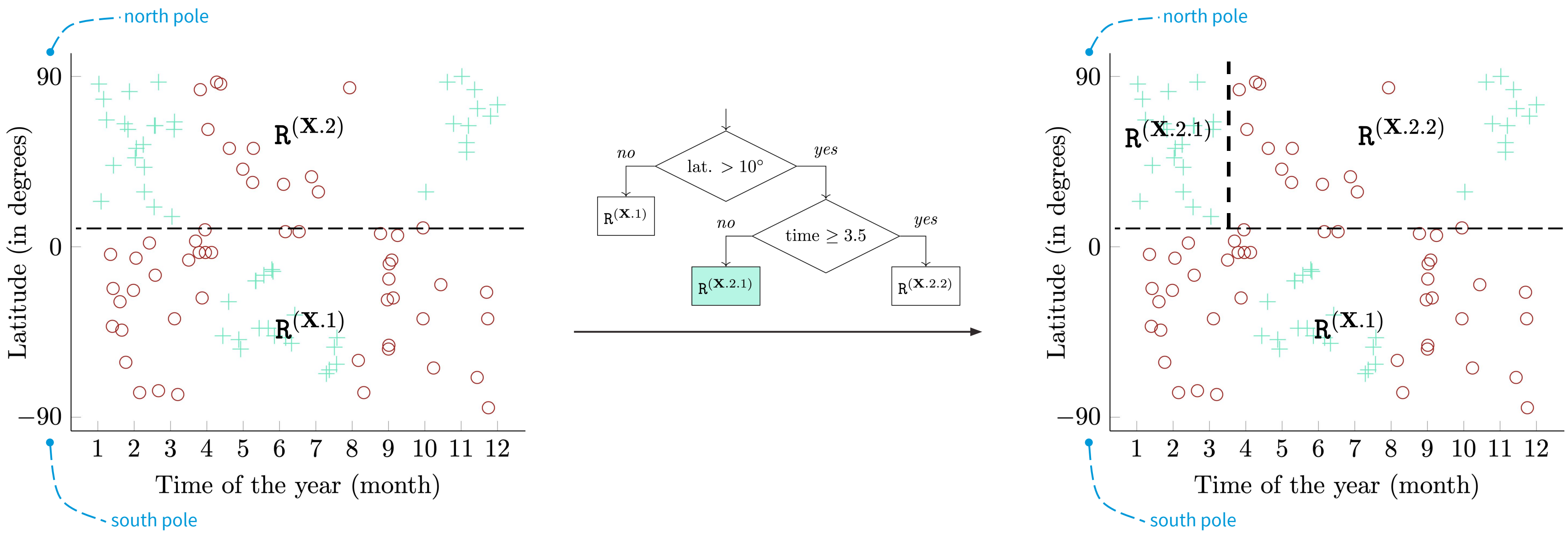




$$\mathbf{X} = \bigcup_{j=1}^{r} \mathtt{R}^{(j)}, \; r \in \mathbb{Z}^{+}$$

(a) Partition the input region $\mathbf{X}$, with a threshold on the latitude, at 10°

Choosing Regions: Top-Down, Recursive, Greedy

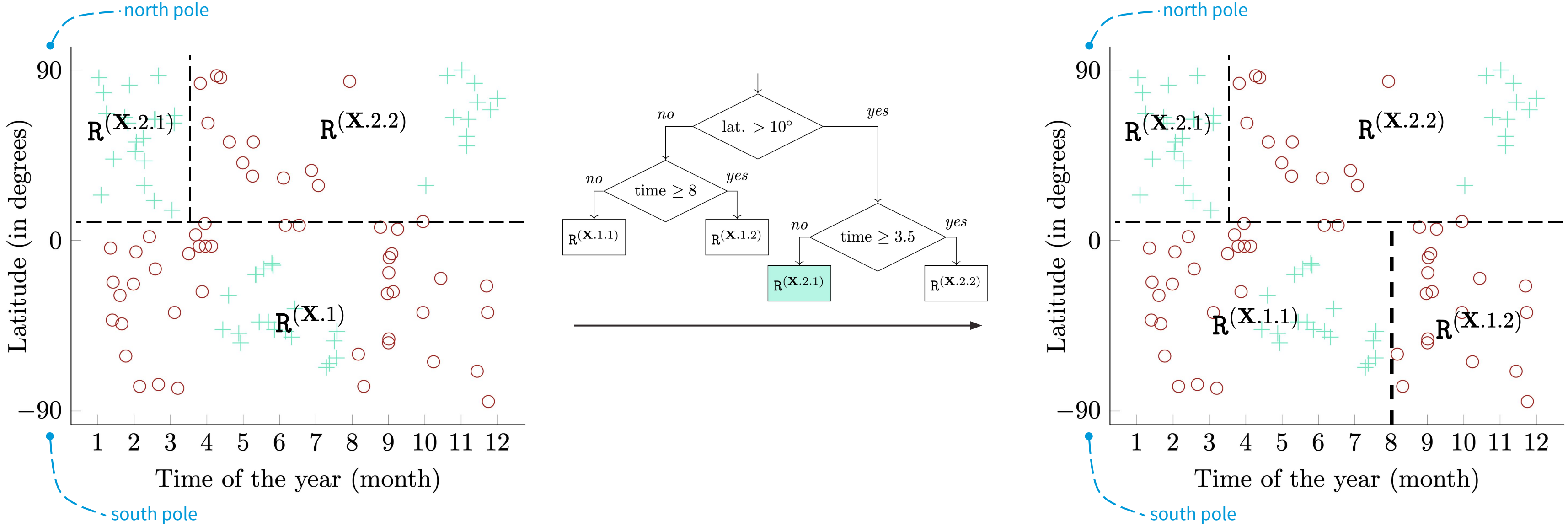


Learning problem: predict whether to ski or not, given the location and time of year

$$\mathbf{X} = \bigcup_{j=1}^{r} \mathtt{R}^{(j)},\ r \in \mathbb{Z}^{+}$$

(b) Partition the top child region of the input space $\mathtt{R}^{(\mathbf{X}.2)}$, with a threshold on time of the year, at 3.5

Choosing Regions: Top-Down, Recursive, Greedy

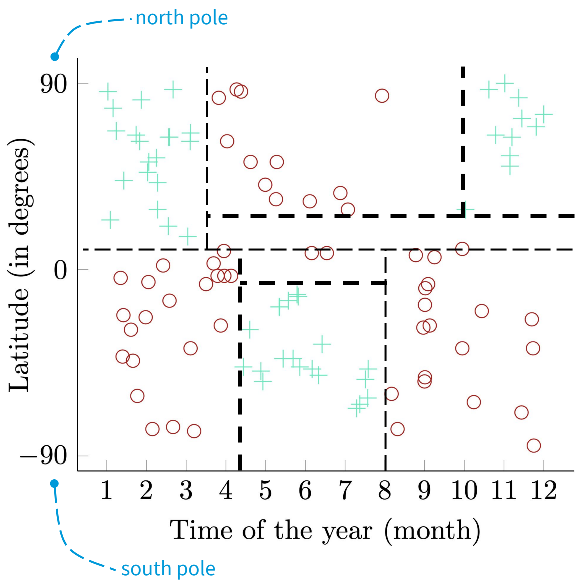


Learning problem: predict whether to ski or not, given the location and time of year

$$\mathbf{X} = \bigcup_{j=1}^{r} \mathtt{R}^{(j)}, \; r \in \mathbb{Z}^{+}$$

(c) Partition the bottom child region of the input space $\mathtt{R}^{(\mathbf{X}.1)}$, with a threshold on time of the year, at 8

Choosing Regions: Top-Down, Recursive, Greedy



Learning problem: predict whether to ski or not, given the location and time of year

$$\mathbf{X} = \bigcup_{j=1}^{r} \mathtt{R}^{(j)},\ r \in \mathbb{Z}^{+}$$

- **Stopping criterion**: purity of leaves (recall the motivation from k–NN); no more attributes left to split[categorical]
  - **Can we get pure leaf nodes for any training data?** — what if we have: $(x^{(i)}, +)$ and $(x^{(j)} = x^{(i)}, -)$?
  - **Majority voting**: if we can **not** grow a consistent tree
- **Why not stop if no split improves the impurity**: greedy algorithm; think of XOR gate!
- **How to split an attribute?**: discrete *vs*. continuous attributes
  - **Discrete**: split falls out naturally (*e.g.*, vampires 🧛)
  - **Continuous**: discretize and find optimal threshold[optim]
- **Loss**: choice of splitting attribute (and threshold)

Q. What happens if we change the stopping condition to include *full growth*?

Agenda

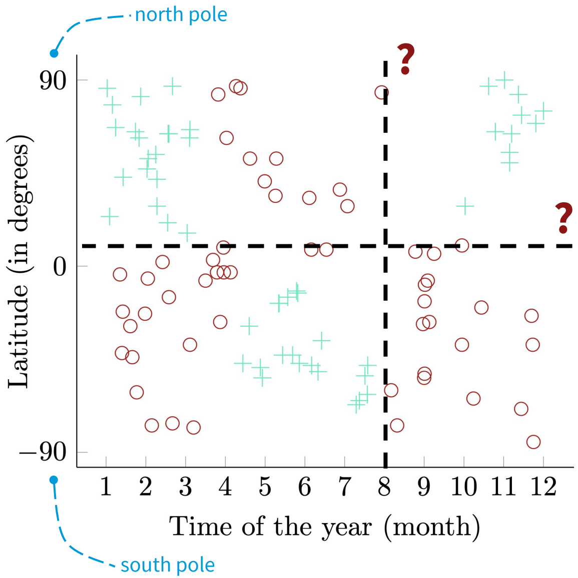
1. Moving From *k*-NN to Decision Trees
2. Region-based Loss Estimation
3. Extensions on Decision Trees
4. Specific Use Cases of Decision Trees
5. Concluding Remarks

Greediness Metric: Information Gain

- **Revisiting Occam's razor**: need to build smallest possible decision tree, greedily
  - **Max decrease in loss**: choose split (s) and threshold (t)
  - **Region-based loss**: cardinality weighted

$$\arg\max_{s,t} \left\{ \underbrace{J(\mathtt{R}^{(\text{parent})})}_{\text{loss at parent node}} - \left(\overbrace{\frac{|\mathtt{R}^{(\text{parent}.1)}|}{|\mathtt{R}^{(\text{parent})}|} J(\mathtt{R}^{(\text{parent}.1)}) + \frac{|\mathtt{R}^{(\text{parent}.2)}|}{|\mathtt{R}^{(\text{parent})}|} J(\mathtt{R}^{(\text{parent}.2)})}^{\text{avg. loss at child nodes}} \right) \right\}$$

- **Information gain**: amount of decrease in loss from parent region to child nodes
  - **Objective**: **maximize** the information gain
  - **Eq.**: minimize the average loss at child nodes
- **Devising loss function**: *pure* = zero loss, *uniform* = max loss



$$\mathbf{X} = \bigcup_{j=1}^{r} \mathtt{R}^{(j)}, \; r \in \mathbb{Z}^{+}$$

Q. Is information gain the best metric of choice? What would happen if we split on *social security number* attribute?

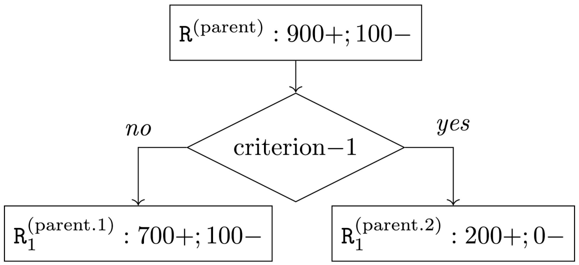
Region-based Loss: Misclassification Error

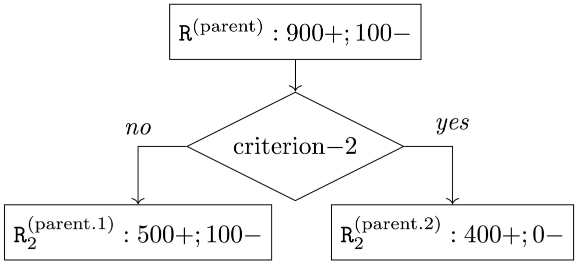
- **Misclassification error by majority vote**: compute the fraction of misclassified samples in a given region
  - **Majority vote**: assume the node label to majority class
  - **Misclassification loss**: $J_{\text{misclass}}(\mathtt{R}^{(j)}) = 1 - \max_{c \in C}(p_c^{(j)})$

    fraction of samples with class 'c' in region 'j'

- **Sensitivity to class probabilities?**: maintaining the misclassification proportion; changing class probabilities
  - Simple loss, meets **greedy** expectations
  - Misclassification loss is quite **insensitive to changes in class probabilities**

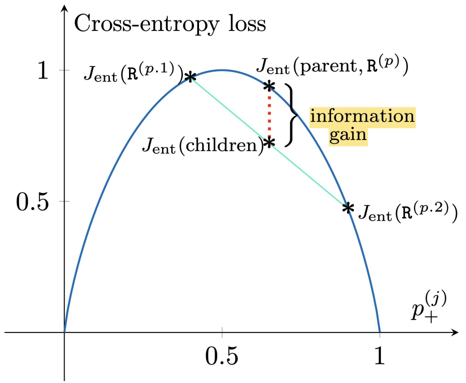
- **Information gain**: whenever the majority vote of the child regions is the same, **zero** information gain!



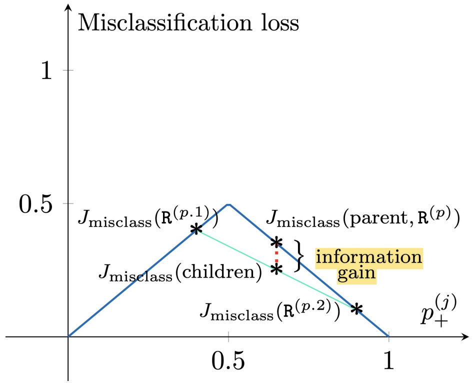


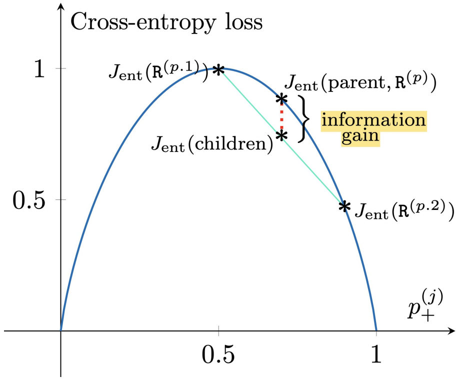
Binary classification misclassification error: $J_{\text{binary-misclass}}(\mathtt{R}^{(j)}) = 1 - \max(p_+^{(j)}, 1 - p_+^{(j)})$

Region-based Loss: Cross-Entropy Loss

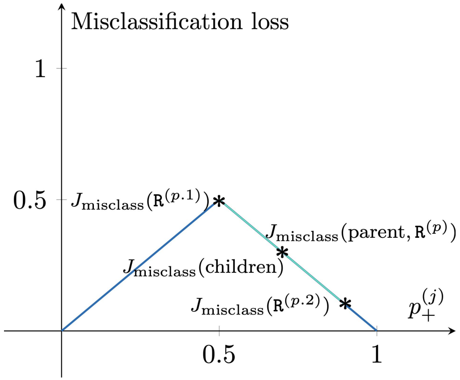


vs.





vs.



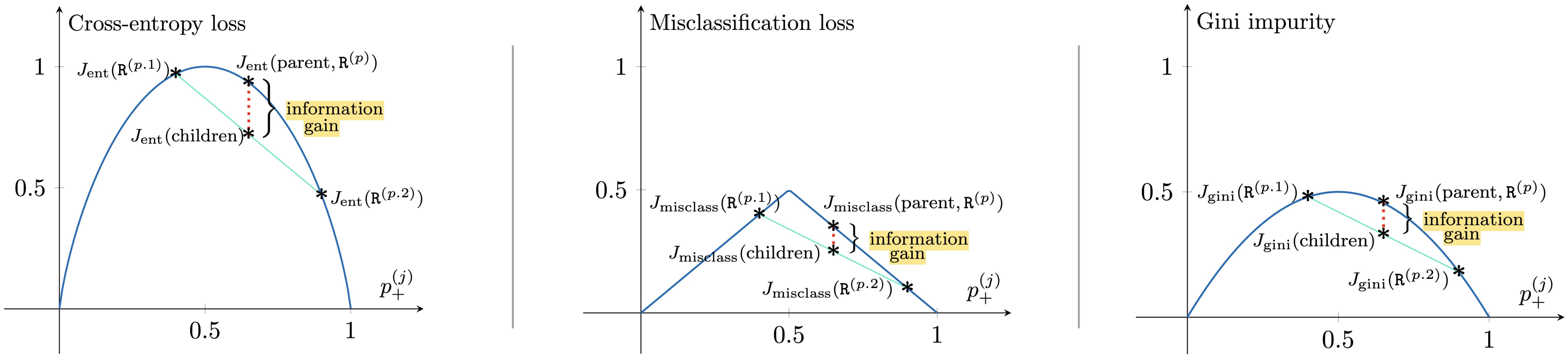
- **Cross-entropy (randomness; disorder)**: measure the level of impurity in a given region — "**strictly concave**"
  - **Information theory**: #bits needed to communicate the class label, given the distribution of proportions
  - **Cross-entropy loss**: $J_{\text{ent}}(\mathtt{R}^{(j)}) = -\sum_{c \in C} p_c^{(j)} \underbrace{\log_2 p_c^{(j)}}$

proportion of samples = 0?

fraction of samples with class '*c*' in region '*j*'

If the base of the logarithm is set to '2,' we have bits/shannons, and if it set to 'e,' we have nats

Region-based Loss: Gini Impurity



- **Gini impurity (misclassification probability)**: choosing a $+$ sample, marking it as $-$; choosing a $-$ sample and marking it as $+$
  - **Gini impurity**: $J_{\text{gini}}(\mathtt{R}^{(j)}) = \sum_{c \in C} p_c^{(j)} \left(1 - p_c^{(j)}\right)$ — fraction of samples with class 'c' in region 'j'
  - **Nature**: Gini impurity is also **strictly concave**, hence, sensitive to class probabilities
  - **Gini impurity vs. cross-entropy**: logarithm approximation (Taylor series + Remez algorithm) is computationally expensive!

Binary classification gini impurity: $J_{\text{binary-gini}}(\mathtt{R}^{(j)}) = 2p_+^{(j)} \left(1 - p_+^{(j)}\right)$

Agenda

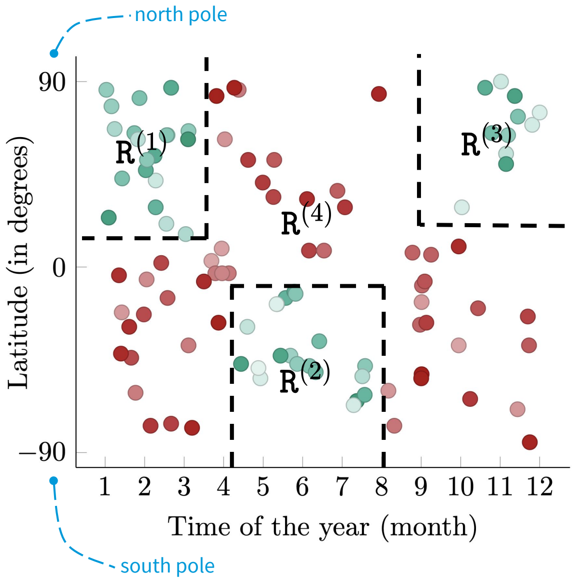
1. Moving From *k*-NN to Decision Trees
2. Region-based Loss Estimation
3. Extensions on Decision Trees
4. Specific Use Cases of Decision Trees
5. Concluding Remarks

Decision Trees for Regression

- **Decision trees for regression**: determine (non-overlapping) areas of interest
  - **Regression**: most tree growth processes remain the same as that employed in classification
  - **Predictions**: rather than majority vote, employ average label, i.e., $y^{(j)} = \frac{1}{|\mathrm{R}^{(j)}|} \sum_{i \in \mathrm{R}^{(j)}} y^{(i)}$
- **Squared loss function**: choose the best attribute, threshold by measuring maximum information gain on squared loss

$$J_{\text{squared}}(\mathrm{R}^{(j)}) = \frac{1}{|\mathrm{R}^{(j)}|} \sum_{i \in \mathrm{R}^{(j)}} \Big(\underbrace{y^{(i)} - y^{(j)}}_{\text{deviation from the average label}}\Big)^2$$

- **Classification And Regression Trees (CART)**: binary trees

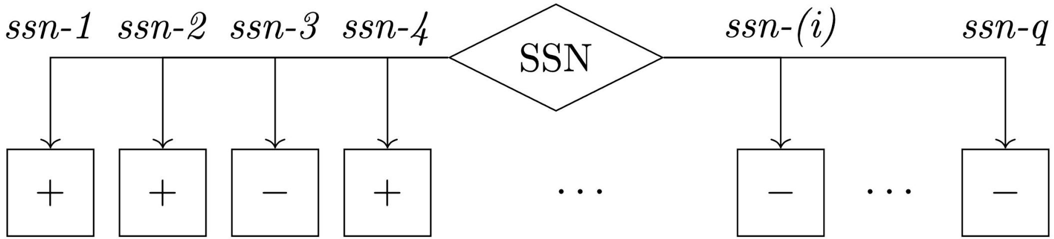


Learning problem: predict the amount of snowfall, given the location and time of year

$$\mathbf{X} = \bigcup_{j=1}^{r} \mathrm{R}^{(j)}, \; r \in \mathbb{Z}^+$$

Leo Breiman, et al. *Classification and regression trees*. International Group, 432:151–166, 1984

Revisiting Categorical Attributes



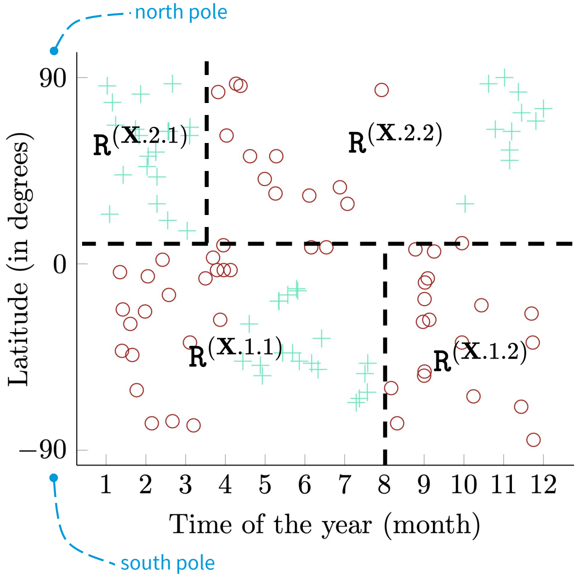
- **Natural threshold**: Categorical attributes, threshold falls out naturally (recall vampires 🧛)
- **Thought experiment**: what would happen if split on highly-branched attributes?

- **Need for 2^q questions**: for a categorical attribute with '*q*' choices, we need 2^q yes-no questions to be answered!
- **Model overfitting**: the use of highly-branching categorical variables to split the input region, often results in high-degree of overfitting
- **Possible solution**: convert highly-branching attributes to numerical attributes: $p^{(j)}_{y^{(i)}} \forall i = 1, 2, \ldots, m$

Another possible solution (used in C4.5 algorithm) is to use *gain ratio*: considers intrinsic information; includes branching details (see §5.3 in lecture notes)

Regularization of Decision Trees

- **Decision trees**: low bias, high variance models — think of the case of full tree growth!
  - **Threshold on the leaf size**: stop splitting when the #samples has reached a minimum threshold
  - **Threshold on the number of nodes in the tree**: stop splitting if #leaves has reached a maximum threshold
  - **Enforce a minimum depth of the tree**: decide to split based on the #splits taken to reach the node
- **Misleading heuristic**: threshold on the obtained information gain (gain ratio) after splits — XOR function!
- **Information-gain based regularization**: Build the entire tree while training, prune away while validating



Learning problem: predict whether to ski or not, given the location and time of year

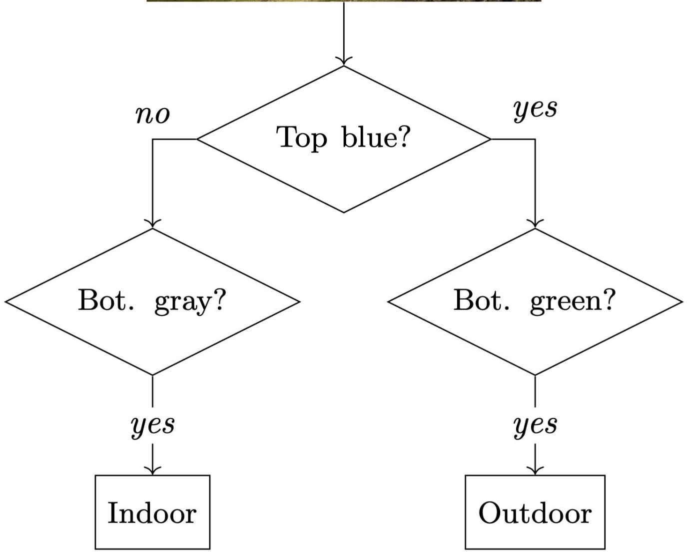
$$\mathbf{X} = \bigcup_{j=1}^{r} R^{(j)},\ r \in \mathbb{Z}^+$$

Information gain and gain ratio are often used together, since gain ratio alone could overcompensate

Agenda

1. Moving From *k*-NN to Decision Trees
2. Region-based Loss Estimation
3. Extensions on Decision Trees
4. Specific Use Cases of Decision Trees
5. Concluding Remarks

Specific Use Cases of Decision Trees



- **Kinect applications**: elementary image classification and posture detection
- **Autopilot**: decision trees had been employed to autopilot an aircraft on a plane simulator by merely learning from the logs of human experts flying the simulator
- **Credit fraud detection**: credit card companies employ decision trees to determine whether or not a loan can be granted to a customer
- **Medical applications**: intuitive to understand and explain, they often mimic the way a doctor thinks, when trained on a medical dataset (caesarean section risk)

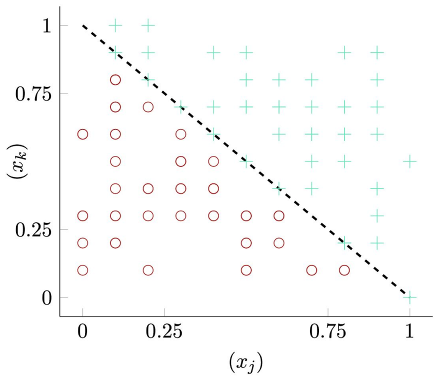
Cynthia J Sims, et al. *Predicting cesarean delivery with decision tree models*. 2000

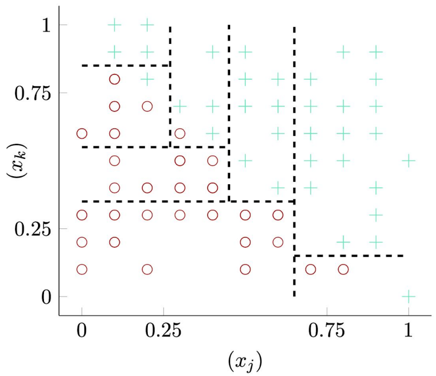
Agenda

1. Moving From *k*-NN to Decision Trees
2. Region-based Loss Estimation
3. Extensions on Decision Trees
4. Specific Use Cases of Decision Trees
5. Concluding Remarks

Concluding Remarks

- **Decision trees**: nonlinear decision boundaries; top-down, recursive, greedy growth
  - **Ease of interpretability and explainability**: a set of if-else rules class, by playing twenty question with data
  - **Speed of training and testing**: small time complexity to train and test decision trees
  - **Support for categorical attributes**: quite easy!
- **Why should you not use decision trees**: basic decision trees model often performs poorly
  - **High model variance**: too much attention to train samples
  - **Lack of additive structure**: linearly separable data
- **Why learn them?**: ideal framework for ensemble learners!





Boosting and bagging approaches: random forests, extra trees, adaboost, gradient boosting have shown promising results on several applications

> Nanos gigantum humeris insidentes

Thank you ~