CS57300
PURDUE UNIVERSITY

NOVEMBER 3, 2021

DATA MINING

ENSEMBLE METHODS

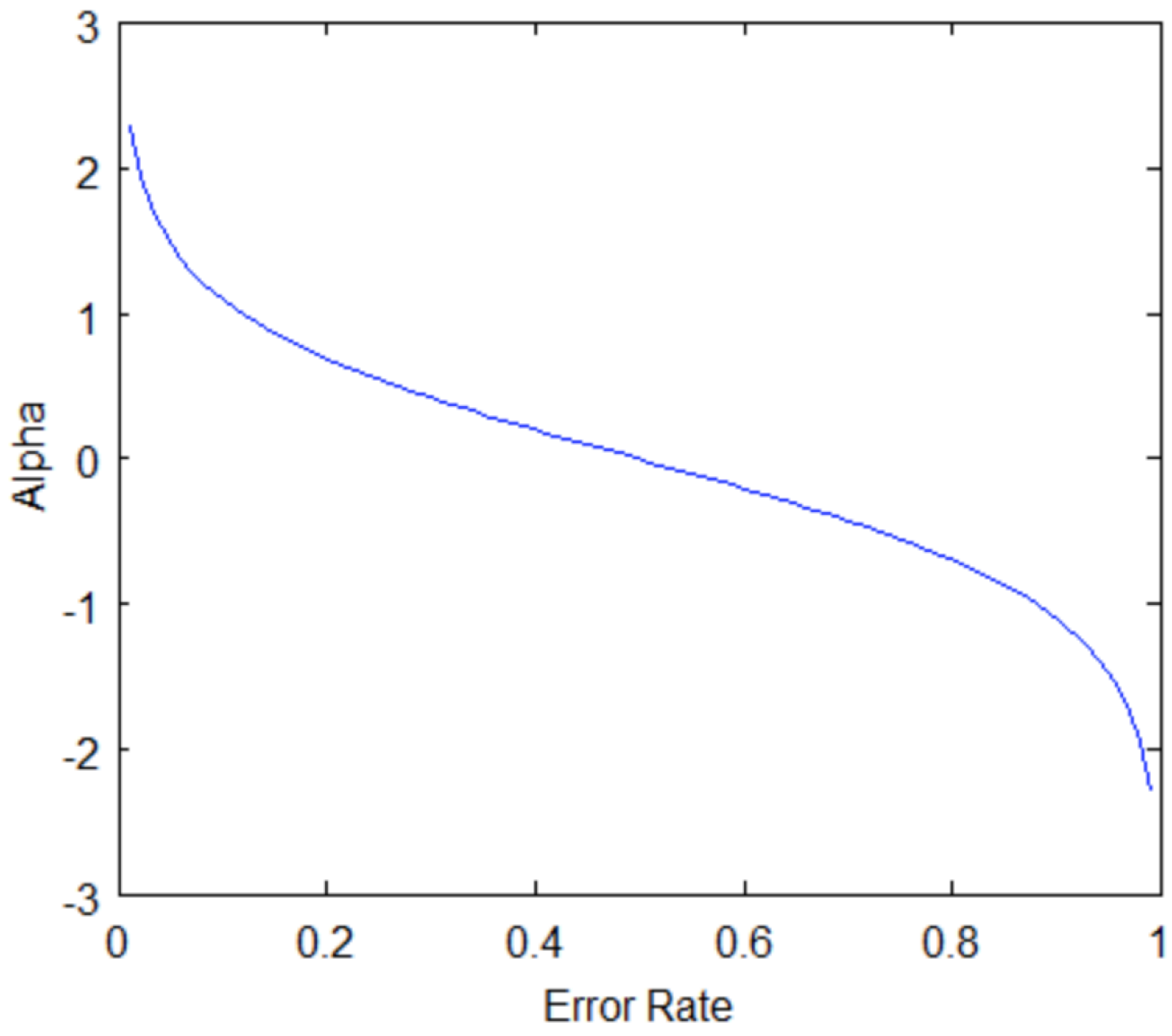
ADABOOST

- Given N training examples $(x_1, y_1), \ldots, (x_N, y_N)$, assign every example in with an equal weight $D_1(i)=1/N$
- For t=1:T
  - Learn model $h_t(x)$ to minimize the weighted error: $\epsilon_t = \mathrm{Pr}_{i \sim D_t}[h_t(x_i) \neq y_i] = \sum_{i=1}^{N} D_t(i) \mathrm{I}(h_t(x_i) \neq y_i)$
  - Set the weight of this model: $\alpha_t = \frac{1}{2} ln(\frac{1-\epsilon_t}{\epsilon_t})$
  - Update training example weights: up-weight the examples that are incorrectly classified and downright examples that are correctly classified: $D_{t+1}(i) = \frac{D_t(i) exp(-\alpha_t y_i h_t(x_i))}{Z_t}$ where $Z_t = \sum_{i=1}^{N} D_t(i) exp(-\alpha_t y_i h_t(x_i))$ is a normalization factor
- To classify new test instance x', apply each model $h_t(x)$ to x' and take weighted vote of predictions

$$H(x') = \mathrm{sign}(\sum_{t=1}^{T} \alpha_t h_t(x'))$$

BOOSTING INTUITION: UNDERSTANDING ALPHA

$$\alpha_t = \frac{1}{2} ln(\frac{1 - \epsilon_t}{\epsilon_t})$$



Low error rate: Large (positive) voting power

Error rate close to 0.5: small voting power

High error rate: Large (negative) voting power

BOOSTING INTUITION: UNDERSTANDING RE-WEIGHTING

$$D_{t+1}(i) = \frac{D_t(i)exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

- When $h_t(x_i) = y_i$, the prediction is correct; $D_{t+1}(i) \propto D_t(i)exp(-\alpha_t)$
- When $h_t(x_i)$!= y_i, the prediction is incorrect; $D_{t+1}(i) \propto D_t(i)exp(\alpha_t)$

WHY ADABOOST WORKS?

- Minimize exponential loss $\sum_{i=1}^{N} exp(-y_i f_T(x_i))$ greedily, where $f_T(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$

WHY ADABOOST WORKS?

- Minimize exponential loss $\sum_{i=1}^{N} exp(-y_i f_T(x_i))$ greedily, where $f_T(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$
- How to get $f_T(x)$ from $f_{T-1}(x)$?

$$\sum_{i=1}^{N} exp(-y_i f_T(x_i)) = \sum_{i=1}^{N} exp(-y_i f_{T-1}(x_i)) exp(-y_i \alpha_T h_T(x_i))$$

$$\propto \sum_{i=1}^{N} D_T(i) exp(-y_i \alpha_T h_T(x_i))$$

$$= \sum_{y_i \neq h_T(x_i)} D_T(i) e^{\alpha_T} + \sum_{y_i = h_T(x_i)} D_T(i) e^{-\alpha_T}$$

$$= \epsilon_T e^{\alpha_T} + (1 - \epsilon_T) e^{-\alpha_T} = \epsilon_T (e^{\alpha_T} - e^{-\alpha_T}) + e^{-\alpha_T}$$

Learn $h_T(x)$ to minimize ϵ_T

Set $\alpha_T = \frac{1}{2} ln(\frac{1 - \epsilon_T}{\epsilon_T})$

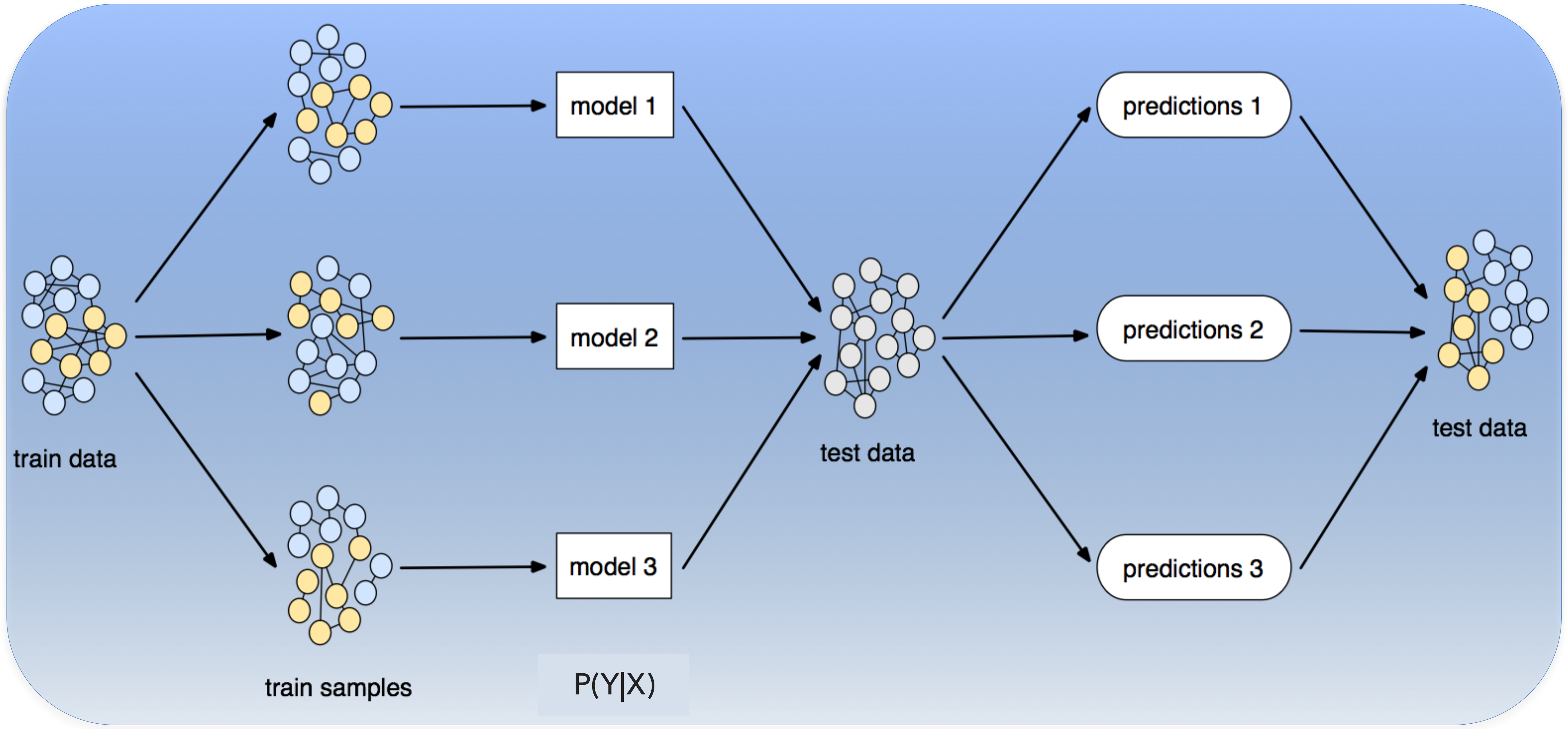
BOOSTING: HOW TO LEARN A MODEL ON WEIGHTED SAMPLES?

- Directly modify the scoring function
  - Weighted log likelihood $\sum_{i=1}^{N} D_t(i) log(\mathsf{P}(y_i | x_i))$ (e.g., logistic regression)
  - Weighted squared loss $\sum_{i=1}^{N} D_t(i)(y_i - o_i)^2$ (e.g, neural network)
- What about models that are learned through heuristic search (e.g., decision trees)?
  - Weighted version of selection criteria: $H(A) = -\sum_{v} wp(x_A = v) log(wp(x_A = v))$, where $wp(x_A = v) = \sum_{i=1}^{N} D_t(i) \mathbb{I}(x_i(A) = v)$
  - Re-sample the training examples according to D_t

BOOSTING

- Main assumption
  - Combining many *weak* (but stable) predictors in an ensemble produces a *strong* predictor (i.e., <u>reduces bias</u>)
  - Weak predictor: only weakly predicts correct class of instances (e.g., decision stumps)
- Model space: non-parametric, can model any function if an appropriate base model is used

BOOSTING



TREATMENT OF INPUT DATA

- re-weight examples

CHOICE OF BASE CLASSIFIER

- weak predictor (e.g., decision stump)

PREDICTION AGGREGATION

- weighted vote

DESCRIPTIVE MODELING

DATA MINING COMPONENTS

- Task specification: **Description**
- Knowledge representation
- Learning technique
- Evaluation and interpretation

DESCRIPTIVE MODELS

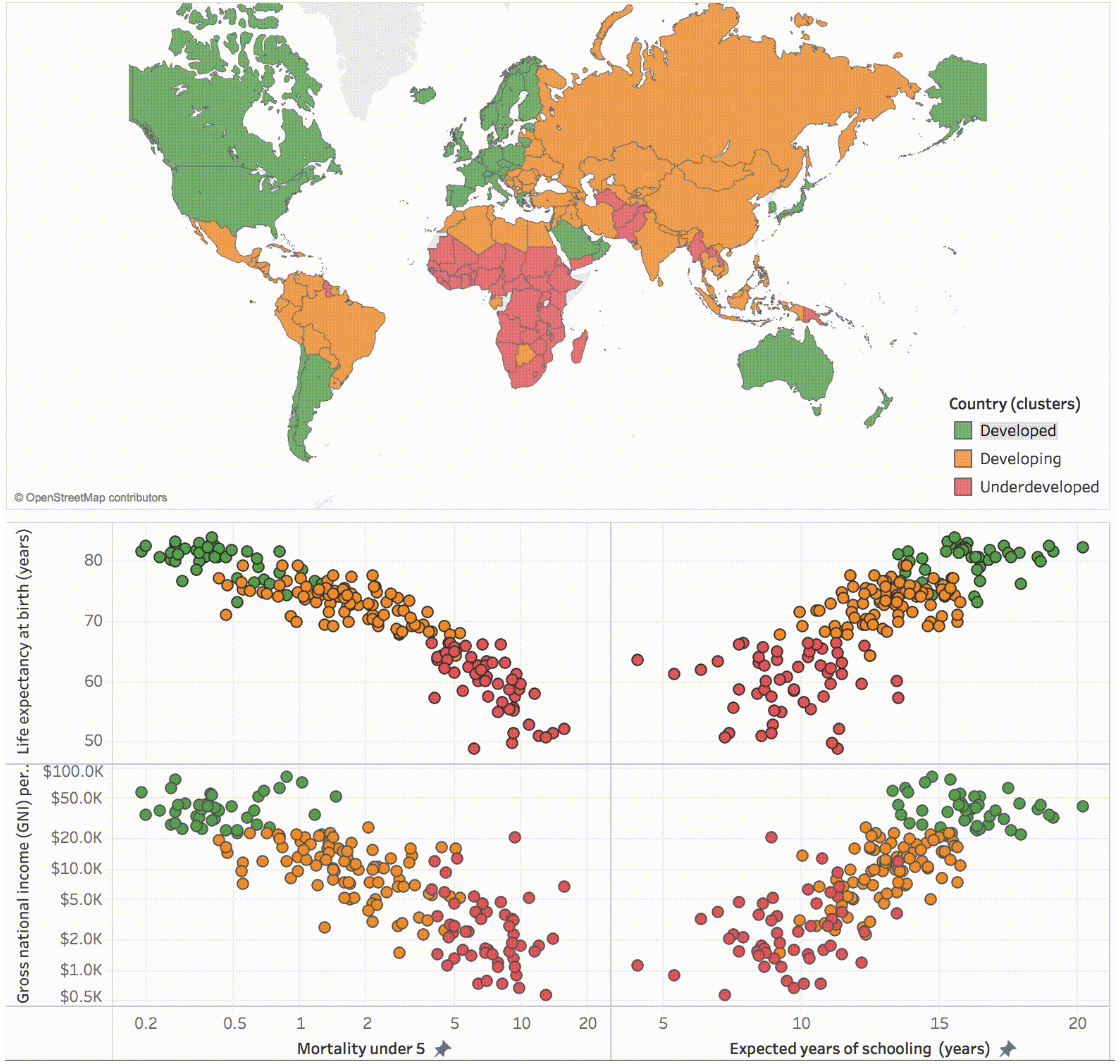
- Descriptive models **summarize** the data
  - Provide a global summary of the data which gives insights into the domain
  - May be used for prediction, but prediction is not the primary goal
- Also known as **unsupervised learning**
  - No predefined “class” labels for each data instance

DESCRIPTIVE MODELING

- Data representation: data instances represented as attribute vectors ***x***(i), often in the form of $n \times p$ tabular data (i.e., p attributes)
- Task–depends on approach
  - Clustering: summarize the data by characterizing groups of similar instances
  - Structure learning and density estimation: determine a compact representation of the full joint distribution $P(\mathbf{X})=P(X_1,X_2,\ldots,X_p)$

CLUSTER ANALYSIS

- Decompose or partition instances into groups s.t.:
  - **Intra**-group similarity is ***high***
  - **Inter**-group similarity is ***low***
- Measure of distance/similarity is crucial

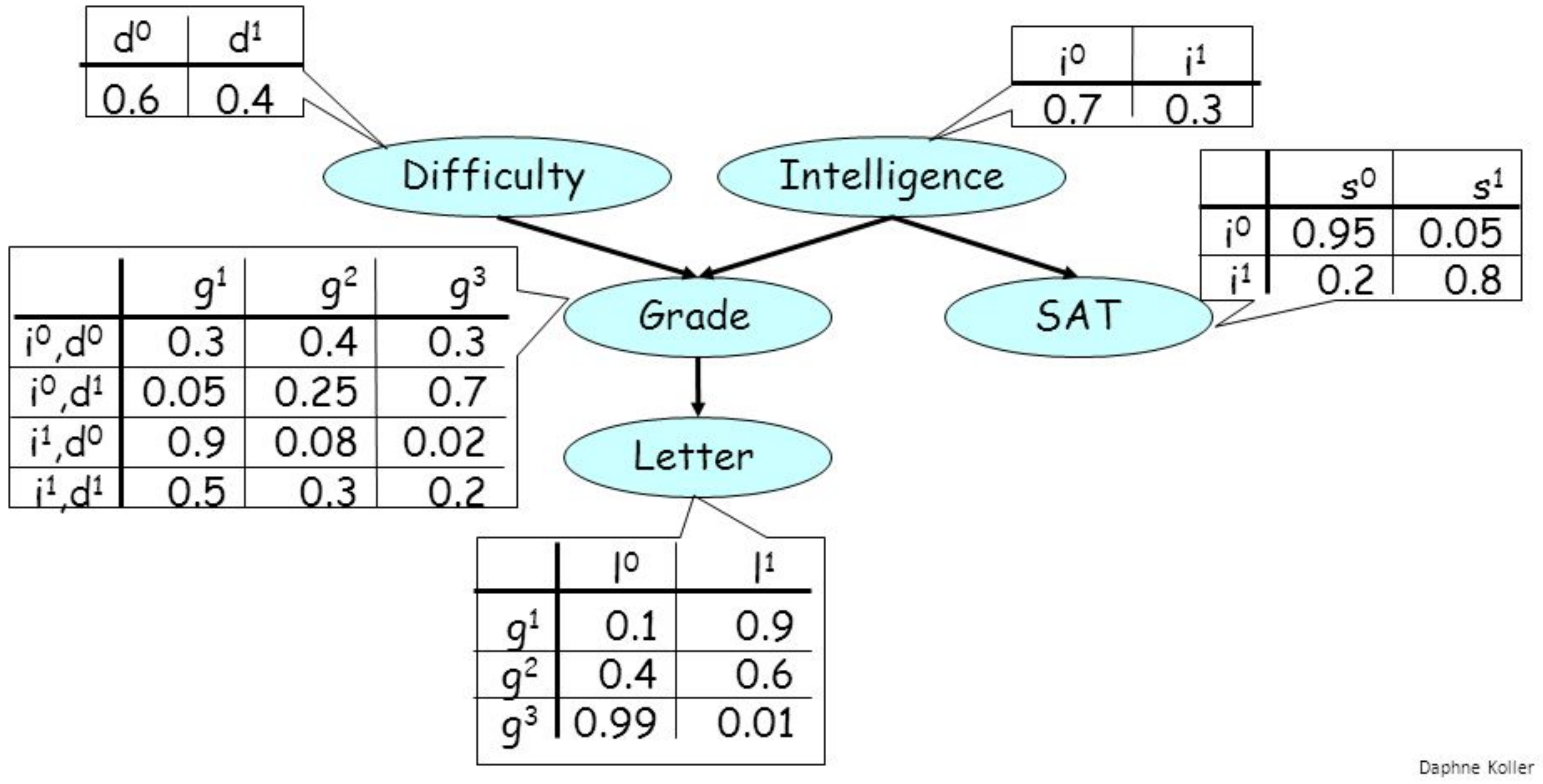


APPLICATION EXAMPLES

- **Marketing**: discover distinct groups in customer base to develop targeted marketing programs
- **Land use**: identify areas of similar use in an earth observation database to understand geographic similarities
- **City-planning**: group houses according to house type, value, and location to identify "neighborhoods"
- **Earth-quake studies**: Group observed earthquakes to see if they cluster along continent faults

STRUCTURE LEARNING AND DENSITY ESTIMATION

- Estimate the structure and parameters for the model that generates the observed data such that:
  - Likelihood of observing the data is high
  - Assumption: data is sampled independently from the same distribution (i.i.d)
- Example
  - Observe data: (student's IQ, student's SAT score, midterm exam difficulty, midterm exam grade, letter quality from the instructor)



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	g^1	g^2	g^3
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^3	0.99	0.01