Decision Trees

Ali Akbar Septiandri

November 17, 2017

untuk Astra Graphia IT

Daftar Isi

Bahan Bacaan

1. VanderPlas, J. (2016). Python Data Science Handbook. O'Reilly Media. https://jakevdp.github.io/PythonDataScienceHandbook/05.08-random-forests.html
2. Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann. (Chapter 6. Trees and rules)
3. Tan, P. N. (2006). Introduction to data mining. Pearson Education India. (Chapter 4. Classification)
4. Besbes, A. (2016, August 10). How to score 0.8134 in Titanic Kaggle Challenge [Blog post]. Retrieved from http://ahmedbesbes.com/how-to-score-08134-in-titanic-kaggle-challenge.html

Pendahuluan

Data Cuaca

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Prediksi apakah John akan bermain tenis

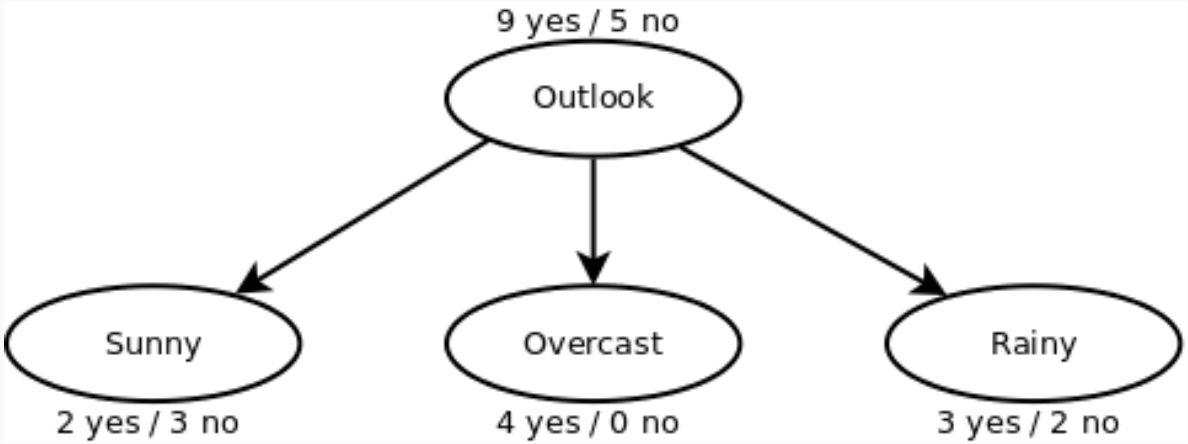
Divide & Conquer

1. Bagi menjadi subsets
2. Apakah pembagiannya murni (semua “ya” atau semua “tidak”)?
3. Jika ya, berhenti
4. Jika tidak, bagi lagi (rekursif)

Data Cuaca

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

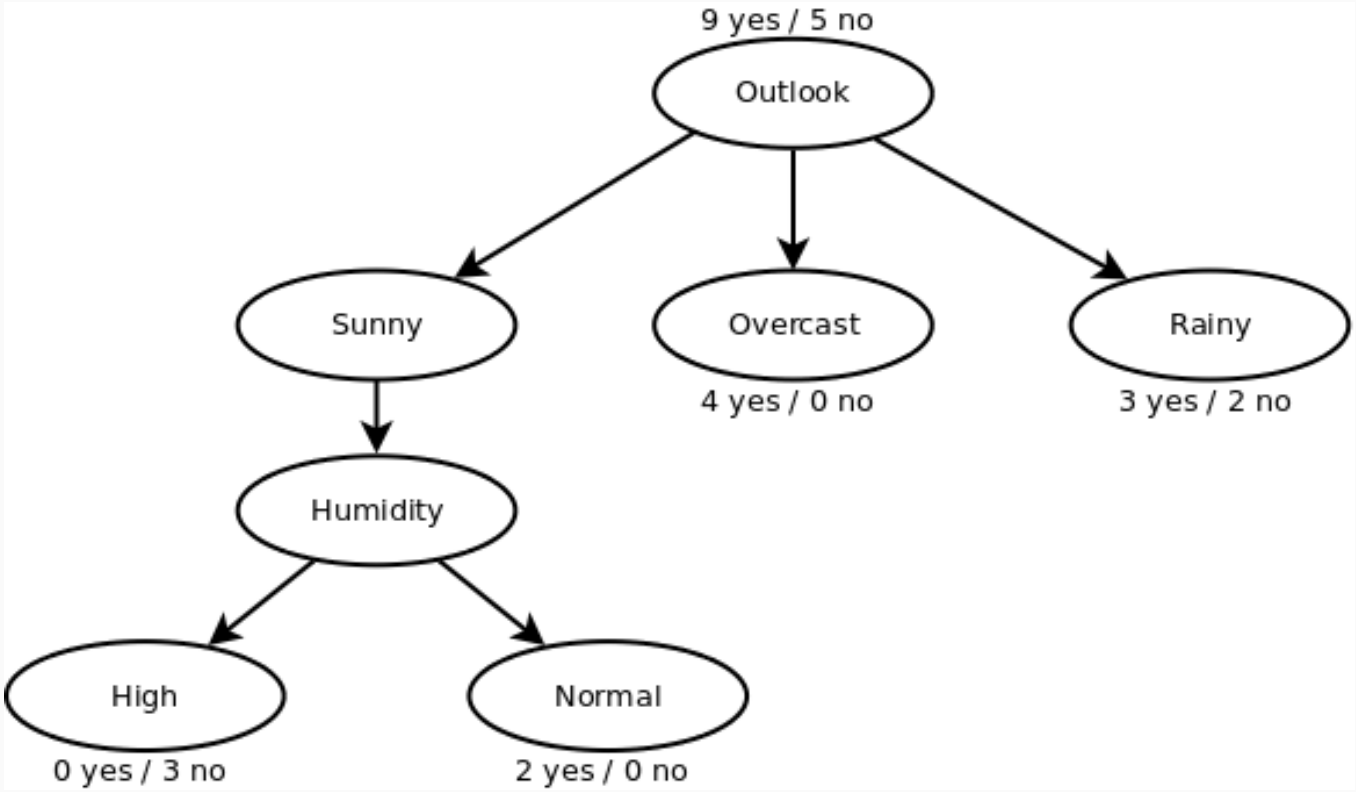
Pohon Keputusan



Data Cuaca

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Sunny	Mild	Normal	True	Yes

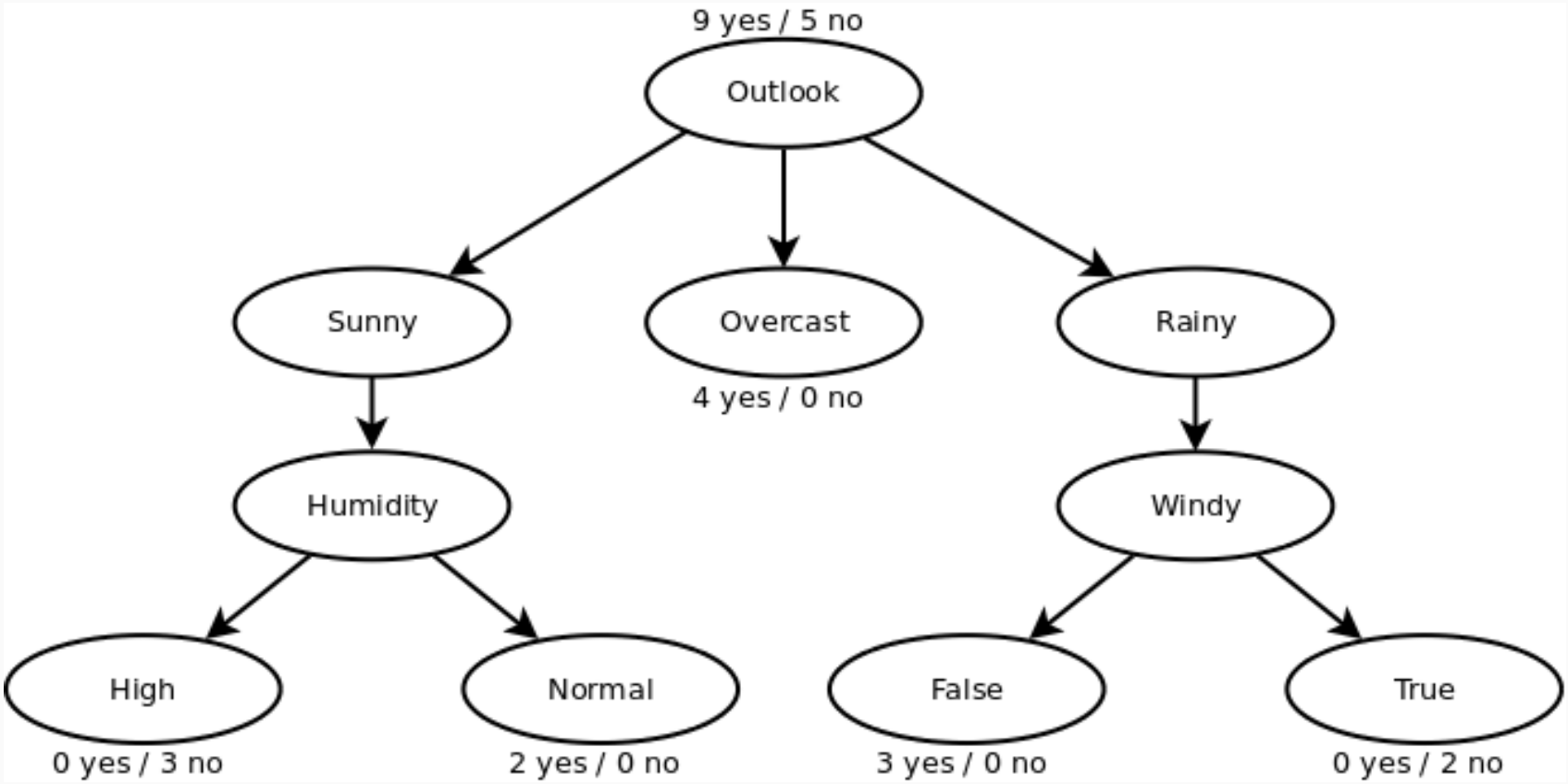
Pohon Keputusan



Data Cuaca

Outlook	Temp	Humidity	Windy	Play
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Rainy	Mild	Normal	False	Yes
Rainy	Mild	High	True	No

Pohon Keputusan



Atribut Pembagi

- Bagaimana menghitung “kemurnian” dari hasil pembagian?
- Bagaimana kalau tidak ada hasil yang langsung murni?
- Atribut mana yang harus didahulukan?

Menghitung Ketakmurnian

Entropy

Formula

$$H(S) = -p_{(+)} log_2 p_{(+)} - p_{(-)} log_2 p_{(-)}$$

dengan S adalah subset dan $p_{(+)}$ dan $p_{(-)}$ adalah persentase (probabilitas) contoh positif atau negatif di subset S

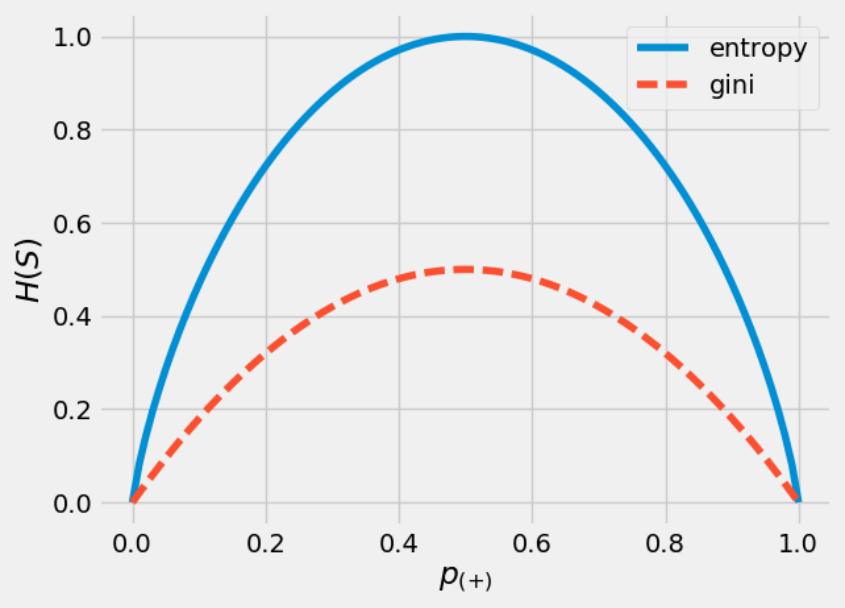
Generalisasi

$$H(S) = -\sum_{c} p_c log_2 p_c$$

Interpretasi

Asumsikan $X \in S$. Berapa *bits* yang dibutuhkan untuk menentukan X bernilai positif atau negatif?

Entropy



Dua contoh kasus:

- Impure (3 yes / 3 no)
  $H(S) = -\frac{3}{6} log_2 \frac{3}{6} - \frac{3}{6} log_2 \frac{3}{6} = 1$
- Pure (4 yes / 0 no)
  $H(S) = -\frac{4}{4} log_2 \frac{4}{4} - \frac{0}{4} log_2 \frac{0}{4} = 0$

Catatan: $0 log_2 0 = 0$ pada perhitungan entropy

Gini Impurity

Formula

$$Gini(S) = 1 - \sum_{c} p_c^2$$

- Digunakan dalam algoritma *classification and regression tree* (CART)
- Interpretasi: Seberapa sering suatu objek akan salah diklasifikasikan jika dilakukan klasifikasi acak

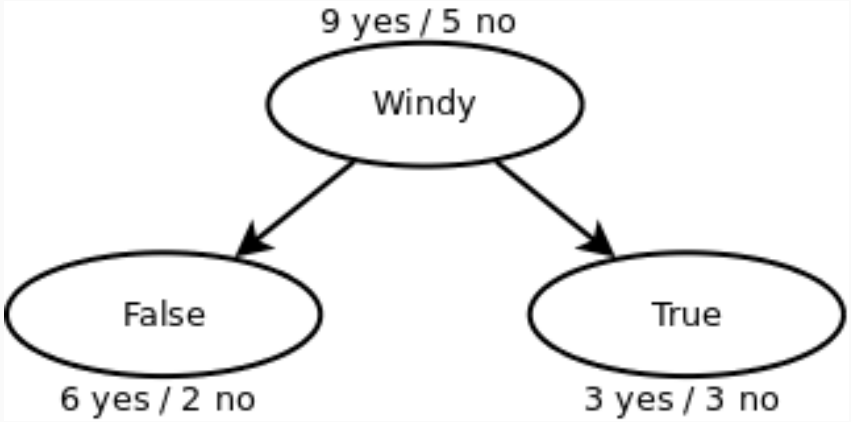
Information Gain

- Kita ingin sebanyak-banyaknya objek dalam *pure sets*
- Melihat perbedaan entropy sebelum dan sesudah dilakukan pemisahan

$$Gain(S, A) = H(S) - \sum_{V \in Values(A)} \frac{|S_V|}{|S|} H(S_V)$$

dengan V adalah nilai yang mungkin dari A dan S_V adalah subset di mana $X_A = V$

Contoh Information Gain



$$H(S) = 0.94, H(S_{False}) = 0.81, H(S_{True}) = 1.0$$

$$Gain(S, Windy) = 0.94 - \frac{8}{14}0.81 - \frac{6}{14}1.0 = 0.049$$

Masalah dengan Information Gain

- Bias terhadap atribut dengan nilai yang banyak
- Tidak dapat berfungsi untuk nilai atribut yang baru
- Solusi: Paksa *binary splits* (CART), atau
- Gunakan GainRatio (C4.5)

$$SplitEntropy(S, A) = -\sum_{V \in Values(A)} \frac{|S_V|}{|S|} log_2 \frac{|S_V|}{|S|}$$

$$GainRatio(S, A) = \frac{Gain(S, A)}{SplitEntropy(S, A)}$$

untuk memberikan penalti untuk atribut dengan nilai yang banyak

Atribut Kontinu

- Intinya, hanya perlu menentukan *threshold*

Atribut Kontinu

- Intinya, hanya perlu menentukan *threshold*
- Masalahnya, perbandingan tiap elemen dengan tiap elemen lainnya akan menghasilkan kompleksitas $O(n^2)$

Atribut Kontinu

- Intinya, hanya perlu menentukan *threshold*
- Masalahnya, perbandingan tiap elemen dengan tiap elemen lainnya akan menghasilkan kompleksitas $O(n^2)$
- Solusi: Urutkan (kompleksitas $O(n \; log \; n)$), lalu ambil titik tengah antara tiap dua nilai

Evaluasi

Error & Akurasi

Setiap hasil klasifikasi akan menghasilkan suatu *confusion matrix*

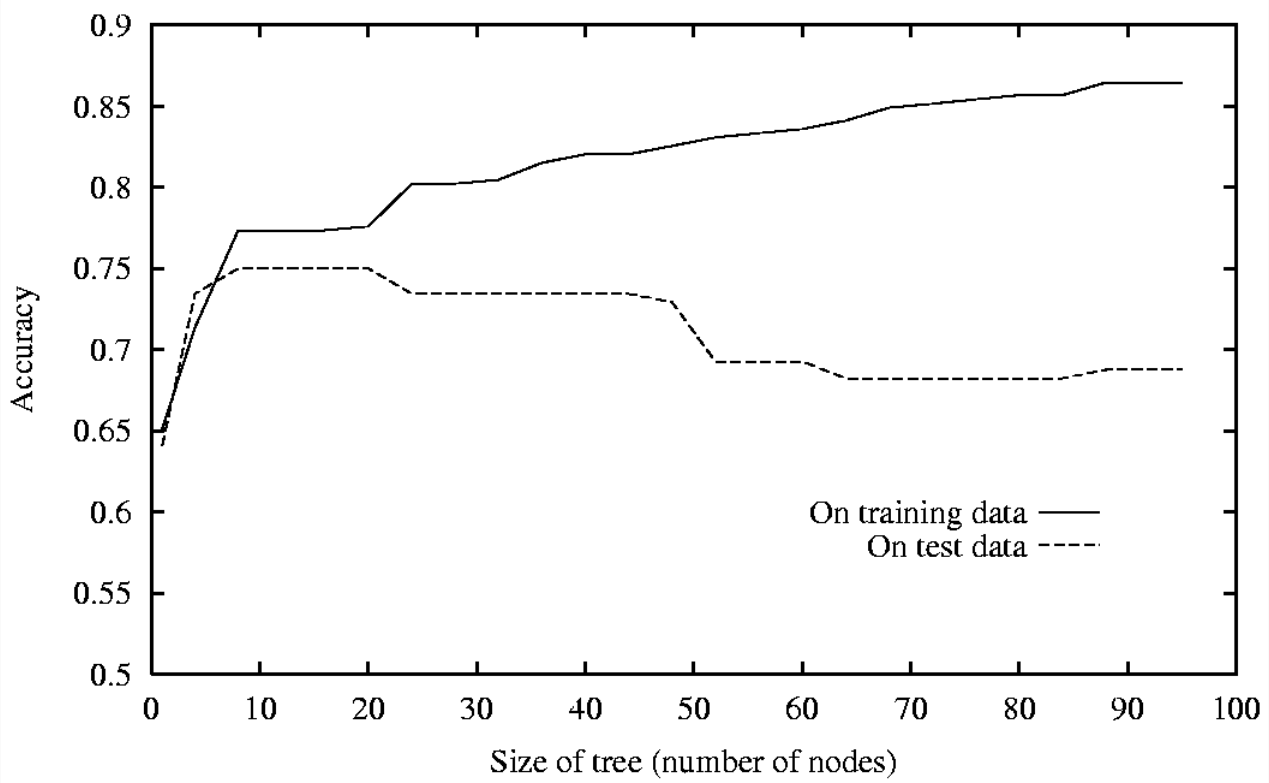
	Ya	**Tidak**
Ya	TP	FN
Tidak	FP	TN

$$Error = \frac{FP+FN}{TP+TN+FP+FN}$$

$$Akurasi = (1 - error) = \frac{TP+TN}{TP+TN+FP+FN}$$

Bagaimana cara meminimalkan error
(memaksimalkan akurasi)?

Overfitting



Gambar 1: Overfitting pada decision trees (Mitchell, 1997)

Menghindari Overfitting

- Hentikan pemisahan saat perubahannya tidak signifikan (*pre-pruning*)

Menghindari Overfitting

- Hentikan pemisahan saat perubahannya tidak signifikan (*pre-pruning*)
- Pisahkan sampai akhir, lalu potong pohonnya (*post-pruning*)
- *Sub-tree replacement pruning* (Witten, et al., 2016; 6.1)

Occam's Razor

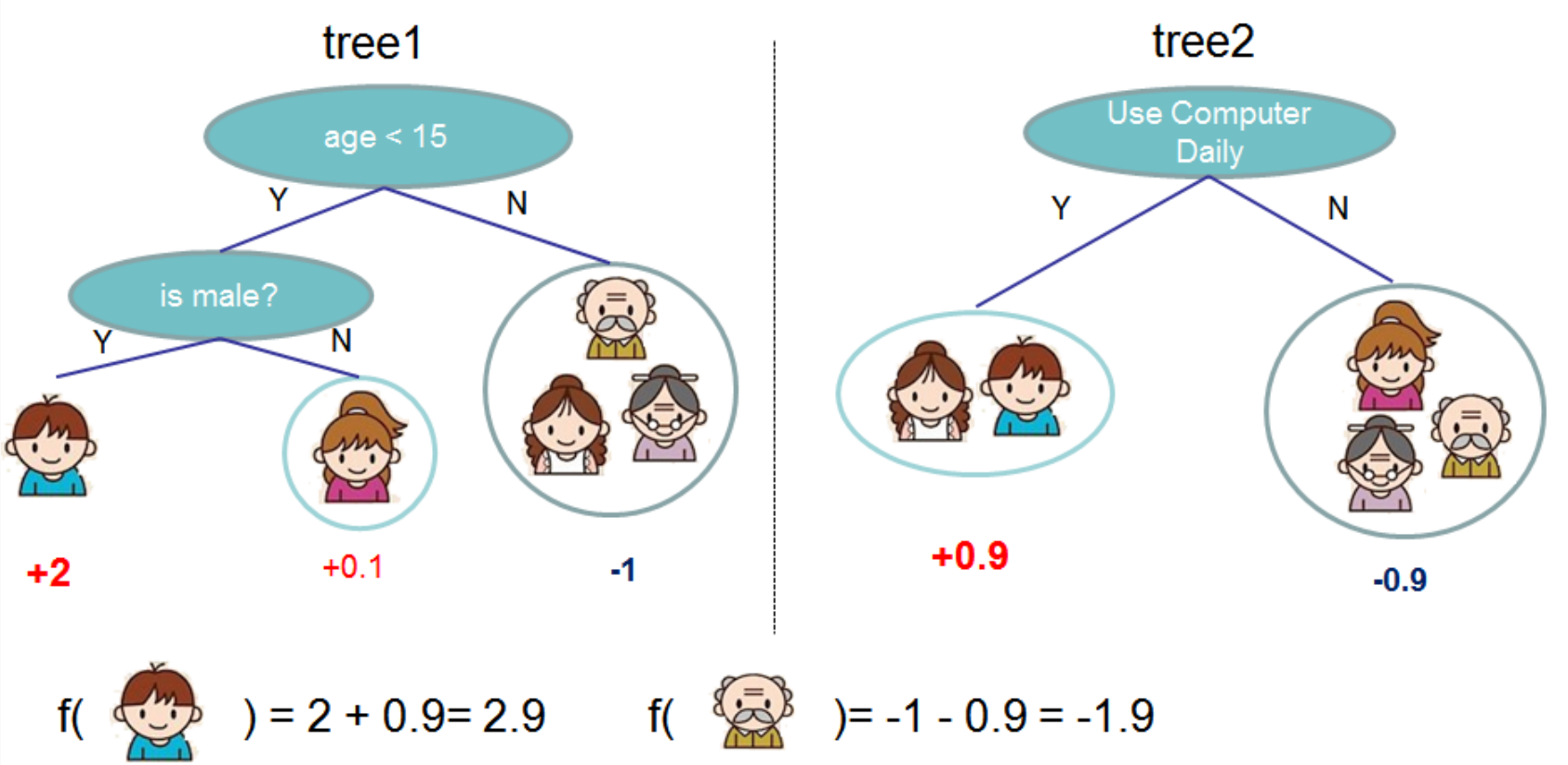
Definisi

Given two models with the same generalization errors, the simpler model is preferred over the more complex model.

Random Forest

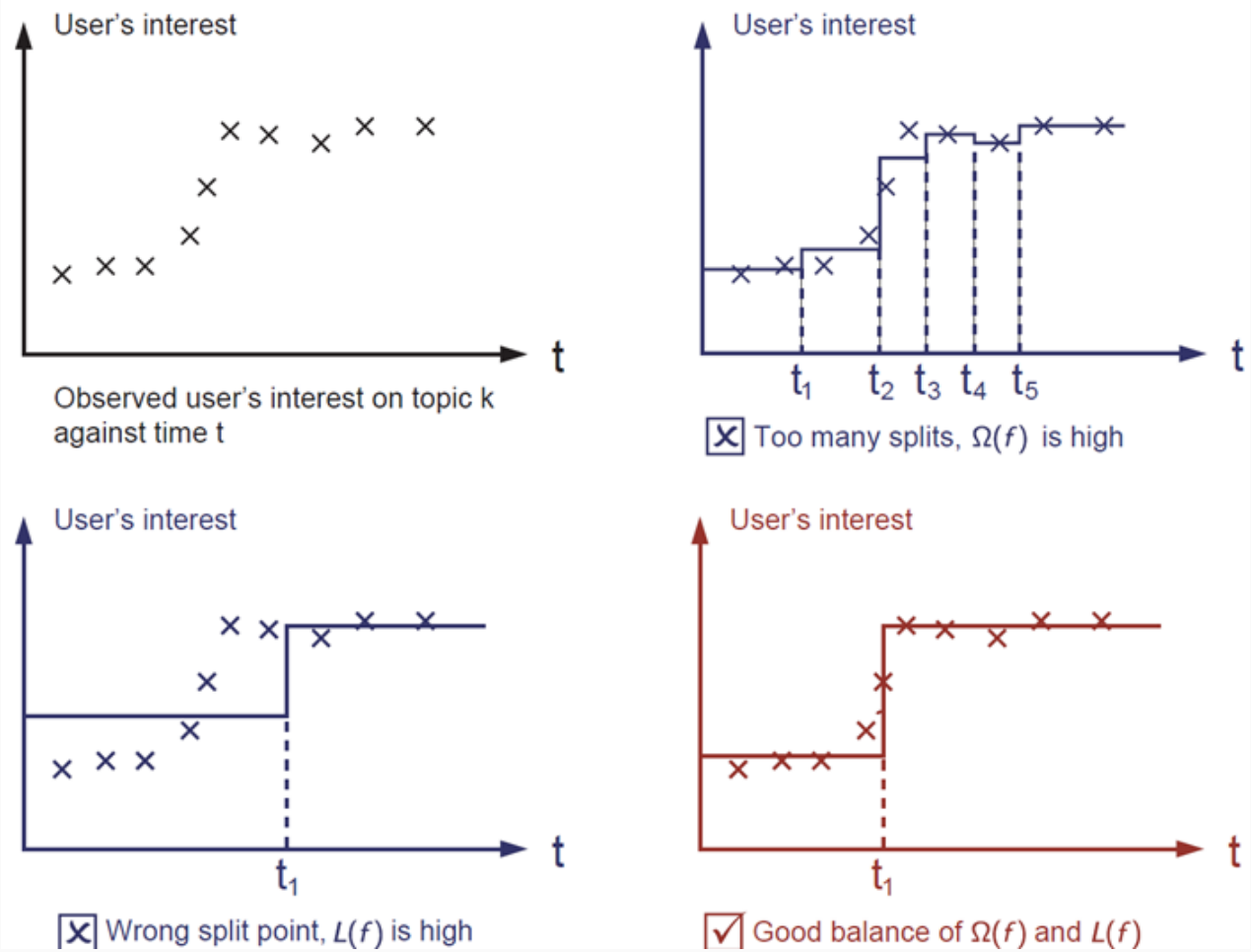
- Membuat K pohon keputusan yang berbeda:
  - memilih subset acak S_r
  - membuat pohon keputusan penuh T_r (tanpa *pruning*)
  - repetisi untuk $r = 1...K$
- Jika diberikan data baru X:
  - klasifikasi dengan setiap pohon $T_1...T_K$
  - Gunakan *majority vote*
  - Alternatif: *weighted average*
- Salah satu metode yang paling efektif (*state-of-the-art*)

Gradient Boosting



Chen, T., & Guestrin, C. (2016, August). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794). ACM.

Regularization



Gambar 2: Regularisasi untuk menghindari *overfitting*

Pros & Cons

Pros

- mudah diinterpretasi
- dapat menangani *missing value*
- sangat cepat saat klasifikasi data baru

Cons

- pembagian hanya sejajar sumbu
- *greedy*, mungkin tidak mencapai solusi optimal global

Terima kasih