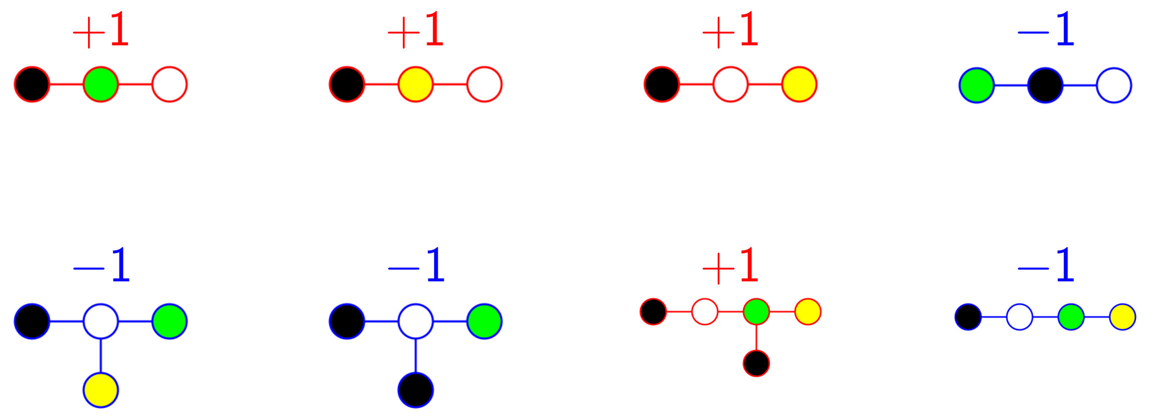
全部分グラフ指示子上の分類森構築に向けて

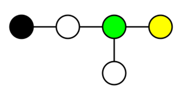
北海道大学大学院 修士1年 横山侑政

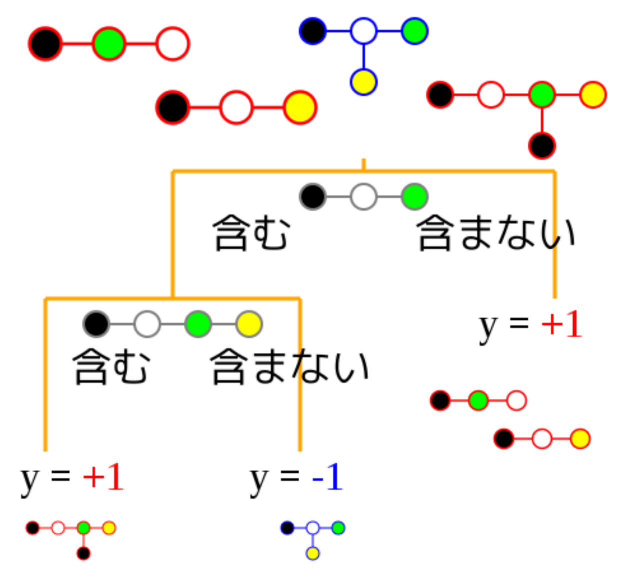
グラフの教師あり分類問題

入力 クラス($y=+1$ or -1)の分かる複数のグラフ N 個



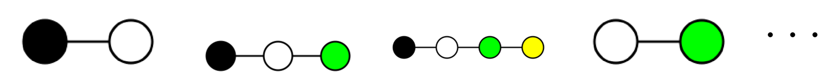
出力 クラスを予測する分類森 F

 $\Rightarrow F \Rightarrow +1$ or -1



$$F_t(x) = \sum_{i=1}^{t} f_i(x) = \sum_{i=1}^{t} \sum_{v \in \text{leaves of } f_i} w_v r_v$$

特徴ベクトル 部分グラフの有無



					…
	0	0	0	1	
	1	0	0	0	
	1	1	0	1	
	1	1	1	1	

全ての部分グラフに関して有無を調べるのは困難
頻出する部分グラフのみ調べる方法もあるが、
可能なら全ての部分グラフを活用したい!
GBDT なら可能だが精度が悪い。XGBoost, RGF でもしたい。

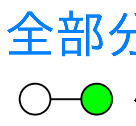
Gradient Boosting Decition Tree

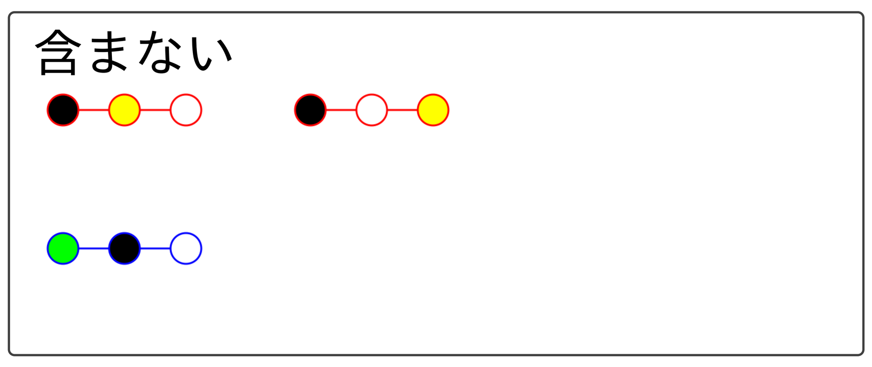
決定木を勾配ブースティングする。正則化を明示的に考えない。
決定木の大きさや縮小率を調整することで暗に正則化している。
明示的な正則化のなければ、全部分グラフ指示子に基づいて学習できる。

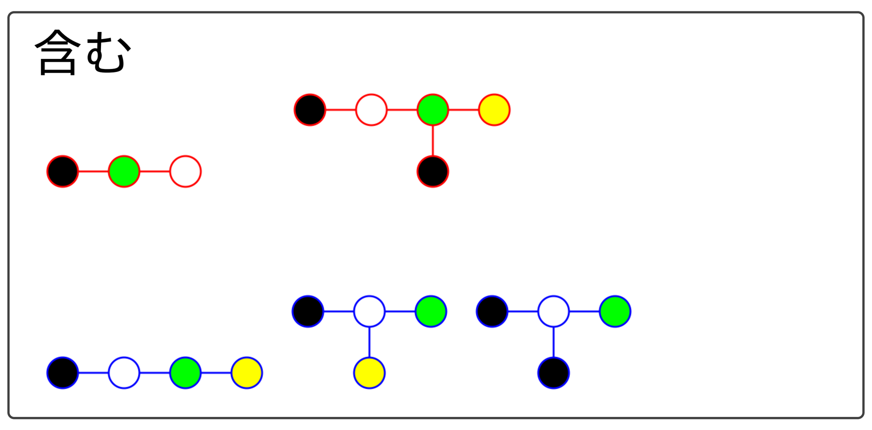
$$\min Q = \sum_{i=1}^{N} \Phi(y_i, F_t(x_i)) = \sum_{i=1}^{N} \Phi(y_i, F_{t-1} + f_t) \approx \sum_{i=1}^{N} \Phi(\tilde{y}_i^{(t-1)}, f_t)$$

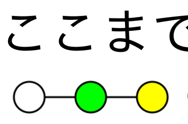
$$\tilde{y}_i^{(t-1)} = -\partial\Phi(y_i, F_{t-t}(x_i)) \ / \ \partial F_{t_1}(x_i)$$

全部分グラフ指示子に基づいた学習

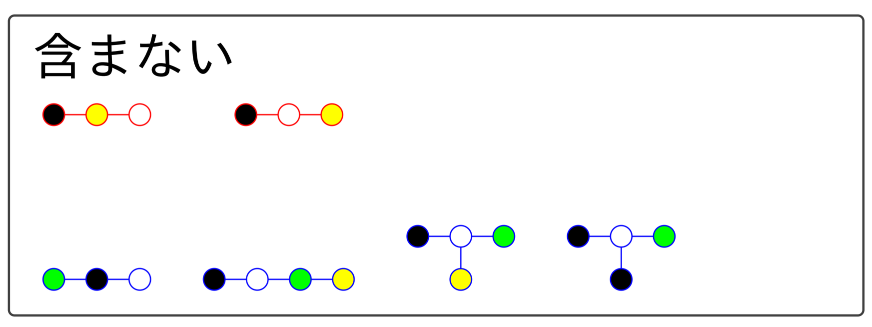
 を

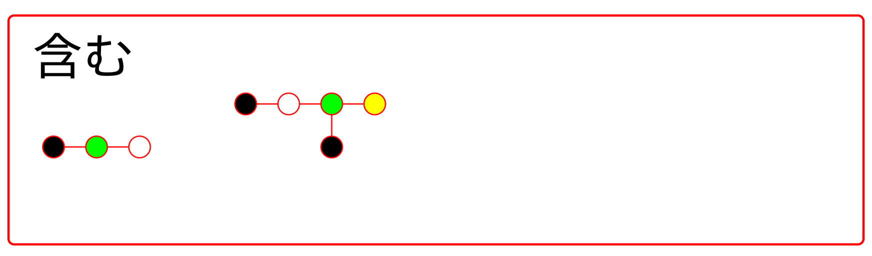




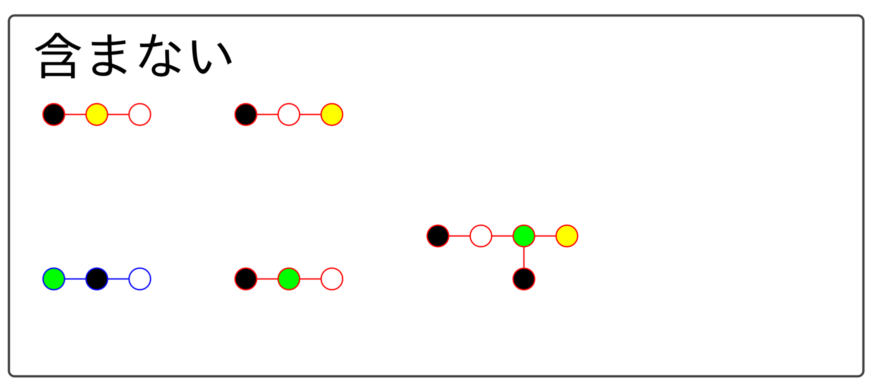
ここまで分かっているとき、
 の有無で分割することを考える

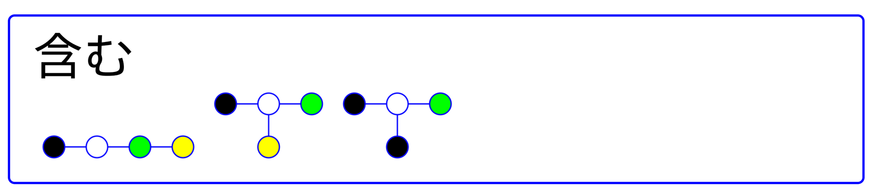
理想の分割($+1$のみ含む)





理想の分割(-1のみ含む)





XGBoost

パッケージ化されており、並列処理などによる高速化がされている。
明示的な正則化があり、目的関数の近似にヘシアンも使う。

$$\begin{aligned}
\min Q &= \sum_{i=1}^{N} \Phi(y_i, F_t(x_i)) + \Omega(F_t) = \sum_{i=1}^{N} \Phi(y_i, F_{t-1} + f_t) + \Omega(f_t) \\
&\approx \sum_{i=1}^{N} [\Phi(\tilde{y}_i^{(t-1)}, y_i) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) \\
&= \sum_{i=1}^{N} [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \\
&= \sum_{v \in \text{leaves of } f_t} [(\sum_{i \in r_v} g_i) w_j + \frac{1}{2}(\sum_{i \in r_v} h_i + \lambda) w_j^2] + \gamma T
\end{aligned}$$

$$\begin{aligned}
&\hat{y}_i^{(t-1)} = F_{t_1}(x_i) \\
&g_i = \partial_{\hat{y}^{(t-1)}} \Phi(y_i, \hat{y}_i^{(t-1)}) \\
&h_i = \partial^2_{\hat{y}^{(t-1)}} \Phi(y_i, \hat{y}_i^{(t-1)}) \\
&\Omega(f_t) = \gamma T + \tfrac{1}{2}\lambda \textstyle\sum_{j=1}^{T} w_j^2
\end{aligned}$$

T は f_t の葉ノードの数

f_t を固定すると、最適なw^*が計算できる。

$$w_j^* = -\frac{\sum_{i \in r_v} g_i}{\sum_{i \in r_v} h_i + \lambda}$$

このとき、目的関数は

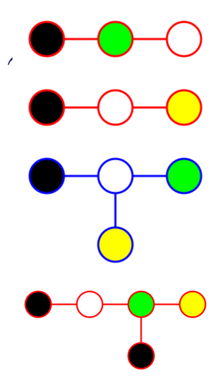
$$\tilde{Q}(f_t) = -\frac{1}{2} \sum_{v \in \text{leaves of } f_t} \frac{(\sum_{i \in r_v} g_i)^2}{\sum_{i \in r_v} h_i + \lambda} + \gamma T$$

f_t の r_0 を r_1 と r_2 に分割するときの目的関数の差分は

$$\frac{1}{2}\left[\frac{(\sum_{i \in r_1} g_i)^2}{\sum_{i \in r_1} h_i + \lambda} + \frac{(\sum_{i \in r_2} g_i)^2}{\sum_{i \in r_2} h_i + \lambda} - \frac{(\sum_{i \in r_0} g_i)^2}{\sum_{i \in r_0} h_i + \lambda}\right] - \gamma$$

分割の目的は

$$\min \left[\frac{(\sum_{i \in r_1} g_i)^2}{\sum_{i \in r_1} h_i + \lambda} + \frac{(\sum_{i \in r_2} g_i)^2}{\sum_{i \in r_2} h_i + \lambda}\right]$$



Regularized Greedy Forest

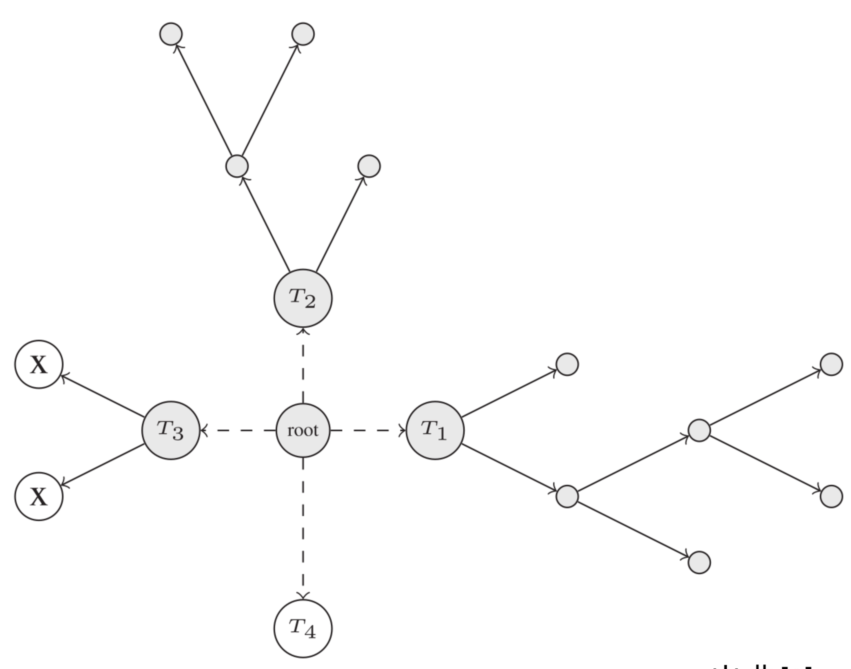
定期的に、全ての葉の重みを計算しなおすことで
収束を早め、簡潔なモデルを作成する。
明示的な正則化があり、勾配ブースティングよりも調整しやすい。
決定木単位ではなく、葉ノード単位でアンサンブルを行う。

$$F(x_i) = \sum_{v \in \text{leaves of } F} w_v r_v$$

r_0 を r_1 と r_2 に分割した分類森 F' は

$$\begin{aligned}
F' &= F - w_0 r_0 + w_1 r_1 + w_2 r_2 \\
&= F - w_0 r_0 + (w_0 + \delta_1) r_1 + (w_0 + \delta_1) r_2 \\
&= F + \delta_1 r_1 + \delta_2 r_2
\end{aligned}$$

$$\hat{\delta}_k = -\frac{Q'(F(\delta_1, \delta_2))}{Q''(F'(\delta_1, \delta_2))} |_{\delta_1=0, \delta_2=0}$$



出典[3]

目的関数 Q は

$$\min Q = \sum_{i=1}^{N} \Phi(y_i, F(x_i)) + \Omega(F) = \sum_{i=1}^{N} \Phi(y_i, F(x_i)) + \lambda \sum_{v} w_v^2/2$$

分割の目的は

$$\begin{aligned}
\min Q' - Q &= \Phi(y_i, w_1 r_1) + \Phi(y_i, w_2 r_2) - \Phi(y_i, w_0 r_0) + w_1^2 + w_2^2 - w_0^2 \\
&= \Phi(y_i, w_1 r_1) + \Phi(y_i, w_2 r_2) + w_1^2 + w_2^2
\end{aligned}$$

参考文献

[1] Friedman, Jerome H. "Stochastic Gradient Boosting." mh (x; am) 1000 (1999): 0.

[2] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." arXiv preprint arXiv:1603.02754 (2016).

[3] Johnson, Rie, and Tong Zhang. "Learning nonlinear functions using regularized greedy forest." IEEE transactions on pattern analysis and machine intelligence 36.5 (2014): 942-954.