TARTU ÜLIKOOL

Loodus- ja täppisteaduste valdkond

Matemaatika ja statistika instituut

Mihkel Jesse

Breimani pügamisteoreemi üldistus

Matemaatika eriala

Bakalaureusetöö (9 EAP)

Juhendajad: prof. Jüri Lember

PhD Kaur Alasoo

PhD Ago-Erik Riet

Tartu 2025

BREIMANI PÜGAMISTEOREEMI ÜLDISTUS

Bakalaureusetöö

Mihkel Jesse

Lühikokkuvõte

Klassifikatsiooni- ja regressioonipuud (CART-puud) on masinõppemeetod. Käesolevas bakalaureusetöös käsitletakse CART-puude kasvatamise, pügamise ja rakendamisega seonduvat teooriat ning üldistatakse Breimani pügamisteoreemi, lisades uudse karistusliikme. Tavaliste CART-puude puhul kasutatakse riskiliiget tükeldamisotsuse hindamiseks, pakutud karistusliikme abil hoitakse tükeldamisel sarnaseid elemente koos. See lähenemine võimaldab CART-puude rakendamist ka klasterdusülesannetes.

CERCS teaduseriala: P160 Statistika, operatsioonianalüüs, programmeerimine, finants- ja kindlustusmatemaatika.

Märksõnad: masinõpe, klassifikatsioon, regressioonanalüüs, puud, klasteranalüüs.

GENERALIZATION OF BREIMAN'S PRUNING THEOREM

Bachelor thesis

Mihkel Jesse

Abstract

Classification and Regression Trees (CART) are a machine learning method. This bachelor's thesis discusses the theory related to growing, pruning, and applying CART, and generalizes Breiman's pruning theorem by introducing a novel penalty term. While conventional CART methods use a risk term to guide splitting decisions, the proposed penalty term helps to keep similar elements together. This approach extends the application of CART to clustering tasks.

CERCS research specialisation: P160 Statistics, operations research, programming,

financial and actuarial mathematics.

Key Words: machine learning, classification, regression analysis, trees, cluster analysis.

Sisukord

Sissejuhatus

Käesoleva bakalaureusetöö eesmärk on üldistada klassifikatsiooni- ja regressioonipuude (CART-puude) teooriale keskset Breimani pügamisteoreemi, lisades uudse karistusliikme. Töös tuginetakse olulisel määral Leo Breimani jt 1984. aastal ilmunud raamatu „Classification and Regression Trees" peatükile „Optimal Pruning".

CART-puud on traditsiooniliselt kasutusel klassifikatsiooni- ja regressiooniülesannetes, kuid antud töö raames kasutatakse meetodit ka ühe klasterdusülesande lahendamiseks. Vajadus tekkis konkreetsest bioloogilisest probleemist, kus eesmärgiks on geneetiliste signaalide jaotamine tükkidesse. Ülesandes on teada vaid signaalide paariviisiline sümmeetriline mõõdik lõigus [0,1]. Karistusliikme kasutamine on antud ülesande lahendamiseks hädavajalik, kuna sellega välditakse lähedaste signaalide määramist erinevatesse tükkidesse.

CART-puud on üks esimesi masinõppemeetodeid. Meetod on tänaseni laialdaselt kasutusel, näiteks meditsiinis, finantssektoris ja bioloogias. CART-puude eelis komplekssemate meetodite ees on nende väga hea tõlgendatavus.

Esimene peatükk annab põhjaliku ülevaate CART-puudega seotud mõistetest, sealhulgas puude kasvatamisest ja pügamisest. Samuti üldistatakse Breimani pügamisteoreemi ja selle eelduseid vastavalt karistusliikme lisamisele.

Teine peatükk käsitleb CART-puude kasutamist klassifikatsiooni- ja regressiooniülesannete lahendamiseks, toetudes professor Jüri Lemberi loodud „Tehisõpe I" konspektile. Kolmandas peatükis lahendatakse praktilise näitena klassikaline masinõppe ülesanne CART-puudega, demonstreerides meetodi toimimist ja selgitades tulemuste analüüsi. Neljandas peatükis käsitletakse bioloogilist kolokalisatsiooni klasterdamisprobleemi, mis oli algseks motivatsiooniks käesoleva töö teema valikul ja karistusliikmete kasutuselevõtul.

1 Klassifikatsiooni- ja regressioonipuude teooria

Käesolev peatükk põhineb valdavalt Leo Breimani jt õpiku „Classification and Regression Trees" 10. peatükil „Optimal pruning" (Breiman jt, 1984). Originaalse panusena oleme lisanud puude riskidele karistusliikme ning vastavalt muudatusele tõestusi kohandanud.

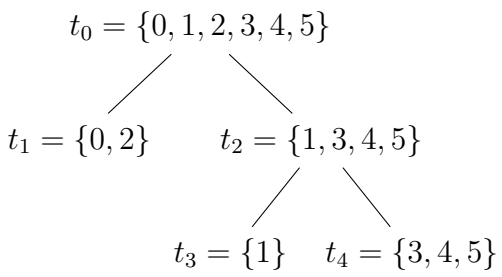
1.1 Mõisted

Selle alapeatüki eesmärgiks on selgitada lahti klassifikatsiooni- ja regressioonipuudega seonduvad mõisted.

Olgu V mittetühi hulk ja E hulga V kõikide kaheelemendiliste alamhulkade hulga alamhulk. Graaf on paar $G = (V, E)$. Hulga V elemente nimetame tippudeks ja hulga E elemente servadeks. Graafi $G = (V, E)$ alamgraafiks nimetame graafi $G' = (V', E')$, kui $V' \subseteq V$ ja $E' \subseteq E$. Ahelaks nimetame tippude $v_1, \ldots, v_r \in V$ järjendit, mille puhul $(v_1, v_2), \ldots, (v_{r-1}, v_r) \in E$. Tsükkel on vähemalt kolme servaga ahel, mis algab ja lõppeb samas tipus ega läbi ühtegi tippu kaks korda. Puu on tsükliteta graaf, mille iga kahe tipu vahel leidub ahel (Laan, 2020, lk 62, 65, 67, 68, 82).

Selles bakalaureusetöös peame edaspidi puude all silmas täis-kahendpuid ehk puid, mille ühel tipul ehk juurel on kaks või null serva ja igal teisel üks või kolm. Kui juurel ei ole ühtegi serva, on terves puus vaid üks tipp ehk tegemist on triviaalse puuga. Kui juurest erineval tipul on kolm serva, siis nimetame seda sisetipuks, kui üks, nimetame seda leheks. Kui puu on triviaalne, siis on juur leht, kui mittetriviaalne, siis sisetipp. Igale tipule vastab mittetühi hulk ning kui selles on vähemalt kaks elementi, siis on meil võimalik seda tükeldada ehk jagada kaheks. Rääkides tipu elementidest peame silmas sellele vastava hulga elemente ning sarnaselt rääkides tipust kui hulgast peame silmas tipule vastavat hulka.

Toome sisse vajalikud tähistused. Olgu T puu, siis tema lehtede hulka tähistame $\tilde{T}$ ning



Joonis 1. Näide täis-kahendpuust.

sisetippude hulka $T - \tilde{T}$. Eeldame, et igal puul on lõplik arv tippe. Olgu t mingi puu T tipp, seda tähistame $t \in T$. Kui $t_1, \ldots, t_n$ on puu T kõik tipud, võime lihtsustatult võrdsustada puud T tema tippude hulgaga ehk $T = \{t_1, \ldots, t_n\}$, eeldades, et tippudevahelised servad on teada. Kui tipp t on puu T juur, tähistame $t =: t_0 =: \text{juur}(T)$. Kui $t \in T$ ei ole juur, kuid tal on juurega t_0 ühine serv, siis ütleme, et ta on juure laps ning juur on tema vanem ehk $t_0 = \text{vanem}(t)$. Tipu lastele määrame järjekorra ning nimetame neid vastavalt vasakuks ja paremaks lapseks, seda tähistame vastavalt $t = \text{vasak}(t_0)$ või $t = \text{parem}(t_0)$. Sarnaselt defineerime ka iga sisetipu lapsed ja vanemad. Kui meil on kaks tippu $t, t' \in T$ ja mingi $n \in \mathbb{N}$ korral kehtib $t = \text{vanem}^n(t')$, siis ütleme, et t on t' eellane ning t' on t järglane. Puu T tipust t algavaks haruks nimetame puud T_t, mille juureks on tipp t ja mis sisaldab kõiki tipu t järglasi puus T. Puu vasakuks ja paremaks peaharuks nimetame harusid, mis algavad vastavalt juure vasakust ja paremast lapsest, vastavalt $t_v := \text{vasak}(t_0)$ ja $t_p := \text{parem}(t_0)$, ning tähistame $T_v := T_{t_v}$ ja $T_p := T_{t_p}$. Kasutame läbivalt tähistust $T = T_v \sqcup T_p$ ning peame silmas, et kaasneb ka juur ehk $t_0 \in T_v \sqcup T_p$.

Olgu T puu, siis funktsioone $R : T \to \mathbb{R}_0^+$ ja $U : T - \tilde{T} \to \mathbb{R}_0^+$ nimetame vastavalt tippude riskiks ja karistusliikmeks. Tipu $t \in T$ riskiks nimetame mittenegatiivset reaalarvu $R(t)$, mis on nii-öelda hind, et tippu koos hoida. Riski määravad selle tipu elemendid ehk risk on alati sama. Sisetipu $t \in T - \tilde{T}$ karistusliikmeks on mittenegatiivne reaalarv $U(t) = U(\{t_1, t_2\})$, kus t_1, t_2 on t lapsed. Karistusliige on sisuliselt tipu tükeldamise

hind ja olenevalt tipu t võimalikest tükeldustest ehk lastest t_1, t_2 on $U(\{t_1, t_2\})$ erinev.

Varasemalt on lähenetud klassifikatsiooni- ja regressioonipuudele ilma karistusliikmeteta, mistõttu on tihti mõistlik tipud esialgselt lahti tükeldada ning liigseid tükeldusi vähendatakse tagantjärgi. Seega, kui iga sisetipu $t \in T - \tilde{T}$ korral $U(t) = 0$, on tegemist klassikalise juhuga. Karistusliiget ei ole varem analoogilistes töödes kasutatud.

Puu T riskiks $R(T)$ nimetame summat

$$R(T) = \sum_{t \in \tilde{T}} R(t) + \sum_{t \in T - \tilde{T}} U(t).$$

Puu T kaalutud riskiks $\lambda \in \mathbb{R}$ järgi $R_\lambda(T)$ nimetame reaalarvu

$$R_\lambda(T) = R(T) + \lambda |\tilde{T}|.$$

Puu risk ei ole aditiivne ehk kui meil on juurega t_0 mittetriviaalne puu T, mille peaharudeks on T_v ja T_p, siis

$$\begin{aligned}
R(T) &= \sum_{t \in \tilde{T}} R(t) + \sum_{t \in T - \tilde{T}} U(t) \\
&= \sum_{t \in \tilde{T}_v \cup \tilde{T}_p} R(t) + \sum_{t \in (T_v \sqcup T_p) - (\tilde{T}_v \cup \tilde{T}_p)} U(t) \\
&= \sum_{t \in \tilde{T}_v} R(t) + \sum_{t \in T_v - \tilde{T}_v} U(t) + \sum_{t \in \tilde{T}_p} R(t) + \sum_{t \in T_p - \tilde{T}_p} U(t) + U(t_0) \\
&= R(T_v) + R(T_p) + U(t_0) \\
&\geq R(T_v) + R(T_p).
\end{aligned}$$

Kuna puude jaoks defineeritud risk ei ole aditiivne, ei ole ka kaalutud risk aditiivne ehk

$$R_\lambda(T) = R_\lambda(T_v) + R_\lambda(T_p) + U(t_0). \tag{1}$$

1.2 Puu kasvatamine

Selles alapeatükis kirjeldatakse puude kasvatamise meetodit. Selleks on meil tarvis juurele vastavat mittetühja ja lõplikku hulka t_0. Iga hulga $t \subseteq t_0$ korral defineeritud riski $R(t)$ ning ka iga $t_1, t_2 \subset t_0$, kus $t_1 \cap t_2 = \emptyset$ karistusliiget $U(\{t_1, t_2\})$.

Puu kasvatamiseks uurime esmalt tippu t_0. Kui see sisaldab vaid ühte elementi, siis meil ei ole võimalik seda tükeldada ning kasvatamine lõppeb. Kui juur ei sisalda vaid ühte elementi, siis kõikidest võimalikest tipu t_0 tükeldustest t_1, t_2 valime sellise, mille korral $R(t_1) + R(t_2) + U(\{t_1, t_2\})$ on minimaalne. Kui meil on mitu sellist tükeldust, siis võime valida kasvatatava tükelduse mingi muu kriteeriumi abil, näiteks kõige võrdsemate tükkide suurustega või kõige esimesena leitud minimaalse riskiga tükelduse. Kui iga t tükelduse t_1, t_2 korral

$$R(t_0) \leq R(t_1) + R(t_2) + U(\{t_1, t_2\}),$$

siis ei ole ühtegi tükeldust, mis vähendaks puu riski ning meie puuks jääb lihtsalt juur t_0 ehk $T = \{t_0\}$. Kui t_0 parim tükeldus t_1, t_2 vähendab riski ehk

$$R(t_0) > R(t_1) + R(t_2) + U(\{t_1, t_2\}),$$

siis kasvatame juurele t_0 lehed t_1 ja t_2 ning fikseerime $U(t_0) := U(\{t_1, t_2\})$.

Edaspidi toimime rekursiivselt tippude t_1 ja t_2 ning võimalusel ka nende järglaste korral. Ehk tipu t_i korral uurime, kas seda on võimalik tükeldada, kui on, siis leiame t_i tükelduse t_{i1}, t_{i2}, mis minimeerib summat $R(t_{i1}) + R(t_{i2}) + U(\{t_{i1}, t_{i2}\})$. Kui see on väiksem kui $R(t_i)$, siis kasvatame tipule t_i lehed t_{i1} ja t_{i2} ning fikseerime $U(t_i) := U(\{t_{i1}, t_{i2}\})$. Kui meil ei ole enam võimalik ühtegi tippu tükeldada või tükeldades riski vähendada, siis oleme oma puu kasvatanud lõpuni ehk jõudnud puuni T_0. Võime puu T_0 tipud ümber indekseerida.

Igal sammul riski minimeerivat tükeldust ei pruugi alati olla arvutuslikult võimalik leida, kuigi on selge, et me soovime puid konstrueerida võimalikult optimaalselt. Praktikas kasvatatakse puid tihti erinevate heuristikatega.

1.3 Pügatud alampuu

Puu T pügatud alampuuks nimetame selle alamgraafi T', kui see on täis-kahendpuu ja sisaldab T juurt. Seda tähistame $T \succeq T'$. Kui aga $T \neq T'$, siis tähistame $T \succ T'$. Seega kehtib ka seos $T \succeq T$. Puu T optimaalseks alampuuks reaalarvu λ järgi nimetame pügatud alampuud T', kui iga T pügatud alampuu T'' korral $R_\lambda(T') \leq R_\lambda(T'')$ ehk optimaalne alampuu on pügatud alampuu, mis minimeerib puu kaalutud riski λ järgi. Minimaalset kaalutud riski λ järgi üle kõikide pügatud alampuude tähistame

$$R^*_\lambda(T) = \min_{T' \preceq T} R_\lambda(T').$$

Optimaalsete alampuude hulka ehk minimaalse riskiga pügatud alampuude hulka tähistame

$$\mathcal{T}_\lambda(T) = \{T' \preceq T : R_\lambda(T') = R^*_\lambda(T)\}.$$

Olgu t_0 mittetriviaalne puu T juur, λ reaalarv, $T'_v \in \mathcal{T}_\lambda(T_v)$, $T'_p \in \mathcal{T}_\lambda(T_p)$ ja $T' = T'_v \sqcup T'_p$, siis seosest (1) järeldub, et

$$R_\lambda(T') = R(T'_v) + R(T'_p) + U(t_0) + \lambda|\tilde{T}'| =$$

$$= R_\lambda(T'_v) + R_\lambda(T'_p) + U(t_0) = R^*_\lambda(T_v) + R^*_\lambda(T_p) + U(t_0)$$

Seega, kui T'' on T mittetriviaalne pügatud alampuu, siis

$$R_\lambda(T'') = R_\lambda(T''_v) + R_\lambda(T''_p) + U(t_0) \geq R^*_\lambda(T_v) + R^*_\lambda(T_p) + U(t_0).$$

Teisalt, kui T'' on triviaalne ehk $T'' = \{t_0\}$, siis $R^*_\lambda(T'') = R(T'') + \lambda = R(t_0) + \lambda$. Kokkuvõttes

$$R^*_\lambda(T) = \min\{R(t_0) + \lambda, R^*_\lambda(T_v) + R^*_\lambda(T_p) + U(t_0)\}.$$

Kuna

$$\mathcal{T}_\lambda(T_v) \sqcup \mathcal{T}_\lambda(T_p) = \{T'_v \sqcup T'_p : T'_v \in \mathcal{T}_\lambda(T_v), T'_p \in \mathcal{T}_\lambda(T_p)\},$$

siis

$$\mathcal{T}_\lambda(T) = \begin{cases} \{t_0\}, & \text{kui } R_\lambda(t_0) < R^*_\lambda(T_v) + R^*_\lambda(T_p) + U(t_0), \\ \mathcal{T}_\lambda(T_v) \sqcup \mathcal{T}_\lambda(T_p), & \text{kui } R_\lambda(t_0) > R^*_\lambda(T_v) + R^*_\lambda(T_p) + U(t_0), \\ \{t_0\} \cup \mathcal{T}_\lambda(T_v) \sqcup \mathcal{T}_\lambda(T_p), & \text{kui } R_\lambda(t_0) = R^*_\lambda(T_v) + R^*_\lambda(T_p) + U(t_0). \end{cases} \quad (2)$$

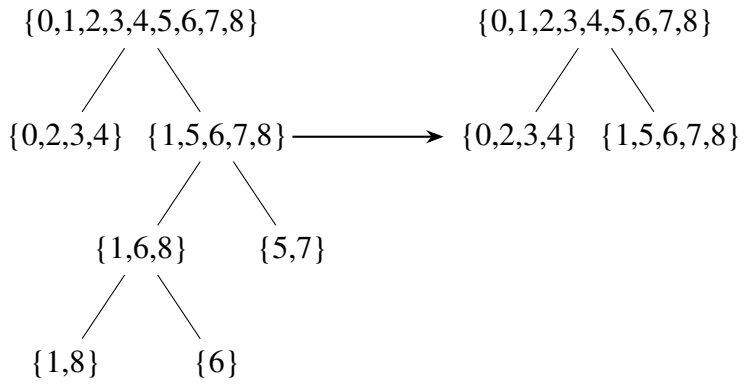
Et optimaalseid alampuid võib olla mitu, siis vähimaks alampuuks nimetame puud $T(\lambda) \in \mathcal{T}_\lambda(T)$, kui iga $T' \in \mathcal{T}_\lambda(T)$ korral $T(\lambda) \preceq T'$. Olgu T_1, T_2 puu T vähimad alampuud $\lambda \in \mathbb{R}$ järgi, siis definitsiooni järgi ühelt poolt iga $T' \in \mathcal{T}_\lambda(T)$ korral $T_1 \preceq T'$, kuna definitsiooni järgi $T_2 \in \mathcal{T}_\lambda(T)$, siis kehtib seos $T_1 \preceq T_2$. Teisalt, iga $T' \in \mathcal{T}_\lambda(T)$ korral kehtib seos $T_2 \preceq T'$ ning et $T_1 \in \mathcal{T}_\lambda(T)$, siis kehtib ka seos $T_2 \preceq T_1$. Kuna mõlemad puud on üksteise pügatud alampuud, siis peab kehtima võrdus $T_1 = T_2$ ehk vähim alampuu, kui eksisteerib, on üheselt määratud.

1.4 Puu pügamine

Selles alajaotuses käsitleme puude pügamise protsessi. Käsitatavaid puid tõlgendatakse eelkõige esialgse hulga tükelduste puudena ja võimalikke lehtede hulki selle mõistlike mitte rangelt binaarsete tükeldustena. Tihti on lõpuni kasvatatud puus liiga palju lehti ehk klastreid, mistõttu ei pruugi need eriti informatiivsed olla. Pügamine on tarvilik, et

puude üldistusvõimekust ja informatiivsust tõsta.

Puu T sisetipu $t \in T$ pügamiseks nimetame sisetipust t algava haru T_t ühendamist tipuks t, mille käigus saame T pügatud alampuu, milles varasem sisetipp t on uue puu leht. Puu T pügamiseks nimetame korduvalt tekkivate alampuude tippude pügamist.



Joonis 2. Tipu pügamine.

Weakest Link Pruning on põhiline CART-puude pügamise meetod. See otsib igal sammul sisetippu, mida pügades kadu suureneks ühendatud lehtede kohta minimaalselt.

Olgu T mittetriviaalne puu. Defineerime funktsiooni $g : \{(t, T) \mid t \in T\} \to \mathbb{R}$ järgmiselt:

$$g(t, T) := \frac{R(t) - R(T_t)}{|\tilde{T}_t| - 1}.$$

Kuna tippe $t \in T - \tilde{T}$ on lõplik arv, siis leidub ka tipp t_1, et $\lambda_1 := g(t_1, T) = \min_{t \in T} g(t, T)$ ja olgu T_1 puu, mille saame esialgsest puust T tippu t_1 pügades. Kui T_1 on mittetriviaalne, leidub sarnaselt ka $\lambda_2 := g(t_2, T_1) = \min_{t \in T_1} g(t, T_1)$ ja T_2. Jätkame sarnaselt pügamist, kuni jõuame puuni $T_r = \{t_0\}$ ja reaalarvuni $\lambda_r = g(t_r, T_{r-1})$.

Oleme saanud pügatud alampuude jada

$$T \succ T_1 \succ \ldots \succ T_r = \{t_0\} \tag{3}$$

ja reaalarvude jada

$$\lambda_1, \lambda_2, \ldots, \lambda_r.$$

Edasi me hakkame sammhaaval tõestama Breimani pügamisteoreemi, mis sisuliselt väidab, et iga reaalarvu λ korral on vähim alampuu mingi puu jadas (3) ja iga $i = 1, \ldots, r$ korral $T_i = T(\lambda_i)$.

Paneme tähele, kui $a, b, c, d, k \in \mathbb{R}$ ning $\frac{a}{b} = \frac{c}{d} = k$, siis $a = kb$ ja $c = kd$, seega

$$\frac{a+c}{b+d} = \frac{kb+kd}{b+d} = \frac{k(b+d)}{b+d} = k$$

ning sarnaselt iga $n \in \mathbb{N}$ korral, olgu $k, a_1, \ldots, a_n, b_1, \ldots, b_n \in \mathbb{R}$, et iga $i \in \{1, \ldots, n\}$ korral $\frac{a_i}{b_i} = k \Leftrightarrow a_i = kb_i$ ehk

$$\frac{\sum_{i=1}^n a_i}{\sum_{i=1}^n b_i} = \frac{\sum_{i=1}^n kb_i}{\sum_{i=1}^n b_i} = \frac{k\sum_{i=1}^n b_i}{\sum_{i=1}^n b_i} = k.$$

Sarnaselt kui meil on n tippu $t_1, \ldots, t_n$, et iga $i \in \{1, \ldots, n\}$ korral $g(t_i, T) = \min_{t \in T} g(t, T) = \lambda$, siis

$$\frac{\sum_{i=1}^n R(t_i) - R(T_{t_i})}{\sum_{i=1}^n (|\tilde{T}_{t_i}| - 1)} = \lambda.$$

Kui meil on pügamise sammul mitu sellist tippu $t_1, \ldots, t_n$, siis pügame need kõik.

1.5 Pügamise omadused

Selles alapeatükis esitame ja tõestame tulemused, mis on vajalikud Breimani pügamisteoreemi tõestuseks. Valdavalt tõestame induktsiooniga lehtede arvu järgi ning baasjuhuna vaatame kui puul on kaks lehte, kuna triviaalse puu korral on kehtivused ilmsed. Kuna puudel on rohkem lehti kui nende peaharudel, saame induktsiooni sammus eeldada,

et väide kehtib peaharude korral. Puude all peame silmas vaid selle peatüki esimeses alapeatükis defineeritud puid. Ei ole tähtis, kuidas puu konstrueeritud on, näiteks võib olla see kasvatatud alapeatükis 1.2 kirjeldatud algoritmiga või mistahes muul viisil.

Lause 1.1 (Breiman jt, 1984, Teoreem 10.7)**.** *Igal puul T leidub vähim alampuu reaalarv λ järgi.*

Tõestus. Olgu T puu. Tõestame induktsiooniga lehtede arvu järgi. Kui $|\tilde{T}| = 2$, siis seosest (2) järeldub, et

$$T(\lambda) = \begin{cases} \{t_0\}, & \text{kui } R_\lambda(t_0) \leq R_\lambda(T), \\ T, & \text{kui } R_\lambda(t_0) > R_\lambda(T) \end{cases} \tag{4}$$

ehk vähim pügatud alampuu tingimata leidub. Kehtigu väide kuni lehtede arvuni $n-1 \geq 2$, näitame, et kehtib ka juhul $|\tilde{T}| = n$. Kuna $|\tilde{T}_v| < n$ ja $|\tilde{T}_p| < n$, siis vastavalt eeldusele leiduvad neil vähimad alampuud $T_v(\lambda)$ ja $T_p(\lambda)$ ning seosest (2)

$$T(\lambda) = \begin{cases} \{t_0\}, & \text{kui } R_\lambda(t_0) \leq R^*_\lambda(T_v) + R^*_\lambda(T_p) + U(t_0), \\ T_v(\lambda) \sqcup T_p(\lambda), & \text{kui } R_\lambda(t_0) > R^*_\lambda(T_v) + R^*_\lambda(T_p) + U(t_0). \end{cases} \tag{5}$$

Seega ka juhul $|\tilde{T}| = n$ leidub vähim alampuu. □

Lause 1.2 (Breiman jt, 1984, Teoreem 10.8)**.** *Olgu T, T_1 puud ja $\lambda \in \mathbb{R}$. Kui $T \succeq T_1 \succeq T(\lambda)$, siis $T(\lambda) = T_1(\lambda)$.*

Tõestus. Eeldusest $T \succeq T_1 \succeq T(\lambda)$ ja vähima alampuu $T(\lambda)$ definitsioonist kehtib võrdus

$$\min_{T' \preceq T} R_\lambda(T') = R_\lambda(T(\lambda)) = \min_{T'_1 \preceq T_1} R_\lambda(T'_1) = R^*_\lambda(T_1).$$

Seega $R^*_\lambda(T_1) = R_\lambda(T(\lambda))$ ja eelduse kohaselt $T_1 \succeq T(\lambda)$, seega vastavalt $T(\lambda)$ definitsioonile $T(\lambda) = T_1(\lambda)$. □

Lause 1.3 (Breiman jt, 1984, Teoreem 10.9)**.** *Olgu T puu, λ_1 ja λ_2 reaalarvud.*

1. *Kui $\lambda_1 \leq \lambda_2$, siis $T(\lambda_1) \succeq T(\lambda_2)$.*

2. *Kui $\lambda_1 < \lambda_2$ ja $T(\lambda_1) \succ T(\lambda_2)$, siis*

$$\lambda_1 < \frac{R(T(\lambda_2)) - R(T(\lambda_1))}{|\tilde{T}(\lambda_1)| - |\tilde{T}(\lambda_2)|} \leq \lambda_2.$$

Tõestus. 1. Kehtigu eeldused. Tõestame induktsiooniga puu T lehtede arvu järgi. Kui $|\tilde{T}| = 2$, siis seosest (4) järeldub kaks võimalust, kas $T(\lambda_1) = \{t_0\}$ või $T(\lambda_1) = T$. Esimesel juhul $R_{\lambda_1}(t_0) \leq R_{\lambda_1}(T)$, seega on vaja näidata, et ka $R_{\lambda_2}(t_0) \leq R_{\lambda_2}(T)$ ehk $T(\lambda_1) = T(\lambda_2) = \{t_0\}$ ja kokkuvõttes $T(\lambda_1) \succeq T(\lambda_2)$. Võrratusest $R_{\lambda_1}(t_0) \leq R_{\lambda_1}(T)$ järeldub, et $0 \leq R(T) - R(t_0) + \lambda_1$ ja kuna $\lambda_1 \leq \lambda_2$, siis ka $0 \leq R(T) - R(t_0) + \lambda_2$ ehk

$$0 \leq R(T) - R(t_0) + 2\lambda_2 - \lambda_2 = R_{\lambda_2}(T) - R_{\lambda_2}(t_0) \Leftrightarrow R_{\lambda_2}(t_0) \leq R_{\lambda_2}(T).$$

Kui $T(\lambda_1) = T$, siis $T \succeq T$ ja $T \succ \{t_0\}$ ehk mõlemad võimalused sobivad.

Kehtigu induktsiooni eeldus kuni $|\tilde{T}| = n - 1$, näitame, et see kehtib ka $|\tilde{T}| = n$ korral. Seosest (5) teame, et on kaks võimalust, kas $T(\lambda_1) = \{t_0\}$ või $T(\lambda_1) = T_v(\lambda_1) \sqcup T_p(\lambda_1)$. Paneme tähele, et vastavalt $T_v(\lambda_1)$ ja $T_p(\lambda_1)$ definitsioonidele ja induktsiooni eeldusele $R_{\lambda_1}(T_v(\lambda_1)) \leq R_{\lambda_1}(T_v(\lambda_2))$ ja $R_{\lambda_1}(T_p(\lambda_1)) \leq R_{\lambda_1}(T_p(\lambda_2))$ ning seega seosest (1) järeldub, et

$$R(T_v(\lambda_1) \sqcup T_p(\lambda_1)) + \lambda_1|\tilde{T}_v(\lambda_1) \cup \tilde{T}_p(\lambda_1)| = R_{\lambda_1}(T_v(\lambda_1)) + R_{\lambda_1}(T_p(\lambda_1)) + U(t_0) \leq$$
$$R_{\lambda_1}(T_v(\lambda_2)) + R_{\lambda_1}(T_p(\lambda_2)) + U(t_0) = R(T_v(\lambda_2) \sqcup T_p(\lambda_2)) + \lambda_1|\tilde{T}_v(\lambda_2) \cup \tilde{T}_p(\lambda_2)|.$$

Esimesel juhul seega

$$\begin{aligned} R(t_0) + \lambda_1 &\leq R(T_v(\lambda_1) \sqcup T_p(\lambda_1)) + \lambda_1 |\tilde{T}_v(\lambda_1) \sqcup \tilde{T}_p(\lambda_1)| \\ &\leq R(T_v(\lambda_2) \sqcup T_p(\lambda_2)) + \lambda_1 |\tilde{T}_v(\lambda_2) \cup \tilde{T}_p(\lambda_2)| \\ &\Updownarrow \\ \frac{R(t_0) - R(T_v(\lambda_2) \sqcup T_p(\lambda_2))}{|\tilde{T}_v(\lambda_2) \cup \tilde{T}_p(\lambda_2)| - 1} &\leq \lambda_1 < \lambda_2 \\ &\Updownarrow \\ R(t_0) + \lambda_2 &< R(\tilde{T}_v(\lambda_2) \cup \tilde{T}_p(\lambda_2)) + \lambda_2 |T_v(\lambda_2) \sqcup T_p(\lambda_2)| \end{aligned}$$

ehk $T(\lambda_2) = \{t_0\}$ ning $T(\lambda_1) = \{t_0\} \succeq \{t_0\} = T(\lambda_2)$. Teisel juhul ehk kui $T(\lambda_1) = T_v(\lambda_1) \sqcup T_p(\lambda_1)$, siis seosest (5), kas $T(\lambda_2) = \{t_0\} \preceq T(\lambda_1)$ või $T(\lambda_2) = T_v(\lambda_2) \sqcup T_p(\lambda_2)$, millest vastavalt induktsiooni eeldusele $T_v(\lambda_1) \succeq T_v(\lambda_2)$ ja $T_p(\lambda_1) \succeq T_p(\lambda_2)$ ning kokkuvõttes $T(\lambda_1) \succeq T(\lambda_2)$.

2. Olgu $\lambda_1 < \lambda_2$ ja $T(\lambda_1) \succ T(\lambda_2)$. Kuna $T(\lambda_1)$ on T vähim alampuu λ_1 järgi ja eeldame, et $T(\lambda_1) \succ T(\lambda_2)$, siis vastavalt vähima alampuu definitsioonile kehtib võrratus $R_{\lambda_1}(T(\lambda_1)) < R_{\lambda_1}(T(\lambda_2))$. Teisalt, et $T(\lambda_2)$ on T vähim alampuu λ_2 järgi, siis kehtib seos $R_{\lambda_2}(T(\lambda_2)) \leq R_{\lambda_2}(T(\lambda_1))$, kuna on ka võimalik, et $T(\lambda_1)$ on T optimaalne alampuu λ_2 järgi, kuid mitte vähim. Seega kehtib seos

$$R_{\lambda_1}(T(\lambda_1)) = R(T(\lambda_1)) + \lambda_1 |\tilde{T}(\lambda_1)| < R(T(\lambda_2)) + \lambda_1 |\tilde{T}(\lambda_2)| = R_{\lambda_1}(T(\lambda_2)),$$

millest omakorda

$$\lambda_1(|\tilde{T}(\lambda_1)| - |\tilde{T}(\lambda_2)|) < R(T(\lambda_2)) - R(T(\lambda_1))$$

ning

$$\lambda_1 < \frac{R(T(\lambda_2)) - R(T(\lambda_1))}{|\tilde{T}(\lambda_1)| - |\tilde{T}(\lambda_2)|}.$$

Teisalt

$$R_{\lambda_2}(T(\lambda_2)) = R(T(\lambda_2)) + \lambda_2|\tilde{T}(\lambda_2)| \leq R(T(\lambda_1)) + \lambda_2|\tilde{T}(\lambda_1)| = R_{\lambda_2}(T(\lambda_1)),$$

millest

$$\frac{R(T(\lambda_2)) - R(T(\lambda_1))}{|\tilde{T}(\lambda_1)| - |\tilde{T}(\lambda_2)|} \leq \lambda_2$$

ehk kokkuvõttes

$$\lambda_1 < \frac{R(T(\lambda_2)) - R(T(\lambda_1))}{|\tilde{T}(\lambda_1)| - |\tilde{T}(\lambda_2)|} \leq \lambda_2.$$

□

Lause 1.4 (Breiman jt, 1984, Teoreem 10.10)**.** *Kui T on puu ja $\lambda \in \mathbb{R}$ ja iga $t \in T - \tilde{T}$ korral $R(t) + \lambda \geq R_\lambda(T_t)$, siis T on üks enda optimaalsetest alampuudest λ järgi ehk*

$$T \in \mathcal{T}_\lambda(T).$$

Tõestus. Tõestame induktsiooniga lehtede arvu järgi. Meenutame, et $T_{t_0} = T$. Kehtigu eeldused ja olgu $|\tilde{T}| = 2$, siis vastavalt eeldusele $R(t_0) + \lambda \geq R_\lambda(T)$ ja seosele (2) kehtib sisalduvus $T \in \mathcal{T}_\lambda(T)$.

Kehtigu väide kuni $|\tilde{T}| = n - 1$ ning näitame kehtivust, kui $|\tilde{T}| = n$. Vastavalt eeldusele $R(t_0) + \lambda \geq R_\lambda(T)$ ja seosele (2) järeldub, et $T_v(\lambda) \sqcup T_p(\lambda) \in \mathcal{T}_\lambda(T)$. Kuna $|\tilde{T}_v| \leq n-1$ ja $|\tilde{T}_p| \leq n - 1$, siis vastavalt induktsiooni eeldusele $T_v \in \mathcal{T}_\lambda(T_v)$ ja $T_p \in \mathcal{T}_\lambda(T_p)$. Seega $R^*_\lambda(T_v) = R_\lambda(T_v)$ ja $R^*_\lambda(T_p) = R_\lambda(T_p)$ ning seosest (1) järeldub

$$R^*_\lambda(T) = R_\lambda(T_v(\lambda) \sqcup T_p(\lambda)) = R_\lambda(T_v(\lambda)) + R_\lambda(T_p(\lambda)) + U(t_0) =$$

$$= R_\lambda(T_v) + R_\lambda(T_p) + U(t_0) = R_\lambda(T)$$

ehk $T \in \mathcal{T}_\lambda(T)$. □

Järgmine lause annab meile eeskirja, mille abil konstrueerida vähimaid alampuid ehk väidab, et vähimas alampuus on vaid tipud, mille iga eellase pügamine suurendaks puu kaalutud riski.

Lause 1.5 (Breiman jt, 1984, Teoreem 10.10)**.** *Olgu T puu, λ reaalarv ja iga $t \in T - \tilde{T}$ korral $R(t) + \lambda \geq R_\lambda(T_t)$, siis*

$$T(\lambda) = \{t \in T : R(s) + \lambda > R_\lambda(T_s) \quad \textit{iga } t \textit{ eellase } s \textit{ korral}\}. \tag{6}$$

Tõestus. Kehtigu eeldused. Tõestame tulemuse induktsiooniga esmalt uurides baasjuhtu $|\tilde{T}| = 2$. Seosest (4) tulenevalt on kaks võimalust: kas $T(\lambda) = \{t_0\}$ või $T(\lambda) = T$. Esimesel juhul $R(t_0)+\lambda = R_\lambda(T)$ ehk $T(\lambda) = \{t_0\}$. Teisel juhul, et $R(t_0)+\lambda > R_\lambda(T)$ ja et $T = T_{t_0}$ ning t_0 on t_v ja t_p eellane, siis $T(\lambda) = \{t_0, t_v, t_p\}$ ehk baasjuhul väide kehtib.

Kehtigu väide kuni $|\tilde{T}| = n - 1$ ning näitame, et see kehtib ka juhul $|\tilde{T}| = n$. Kuna $|\tilde{T}_v| \leq n-1$ ja $|\tilde{T}_p| \leq n-1$, siis vastavalt induktsiooni eeldusele peaharude korral väide kehtib. Samas ka lause 1.4 järgi $R_\lambda(T_v) = R^*_\lambda(T_v)$ ja $R_\lambda(T_p) = R^*_\lambda(T_p)$. Eeldusest on meil kaks võimalust, kas $R(t_0) + \lambda = R_\lambda(T)$ või $R(t_0) + \lambda > R_\lambda(T)$. Kui $R(t_0) + \lambda = R_\lambda(T)$, siis seosest (1)

$$R(t_0) + \lambda = R_\lambda(T) = R_\lambda(T_v) + R_\lambda(T_p) + U(t_0) = R^*_\lambda(T_v) + R^*_\lambda(T_p) + U(T_0).$$

Seosest (5) ja eelnevast võrdusest järeldub, et $T(\lambda) = \{t_0\}$. Kuna t_0 on iga $t \in T \setminus \{t_0\}$ eellane, siis seos (6) kehtib.

Kui $R(t_0) + \lambda > R_\lambda(T)$, siis sarnaselt

$$R(t_0) + \lambda > R_\lambda(T) = R_\lambda(T_v) + R_\lambda(T_p) + U(t_0) = R^*_\lambda(T_v) + R^*_\lambda(T_p) + U(T_0)$$

ning seosest (5) järeldub, et $T(\lambda) = T_v(\lambda) \sqcup T_p(\lambda)$. Kuna t_0 on iga $t \in T_v$ ja $t \in T_p$ eellane, siis koos induktsiooni eelduse rakendamisega puudele $T_v(\lambda)$ ja $T_p(\lambda)$ kehtib seos (6). □

Järgmine järeldus tuletub eelmisest lausest ning on konkreetne viis, kuidas tolle lause abil vähimaid alampuid konstrueerida, mis on kasulik edaspidistes tõestustes ja ka praktikas. Lauset saame korduvalt rakendada, et leida kõik vähimad alampuud. Järeldus kehtib, kuna λ_1 definitsioonist järeldub $R(t) + \lambda_1 \geq R_\lambda(T_t)$ iga $t \in T - \tilde{T}$ korral lause 1.5 tingimuste täitmiseks.

Järeldus 1.6 (Breiman jt, 1984, Teoreem 10.11)**.** *Kui T on mittetriviaalne puu ja $\lambda_1 = \min_{t \in T} g(t, T)$, siis*

$$T_1 = \{t \in T : R(s) + \lambda_1 > R_{\lambda_1}(T_s) \quad \textit{iga } t \textit{ eellase } s \textit{ korral}\}.$$

Lause 1.7 (Breiman jt, 1984, Teoreem 10.11)**.** *Olgu T puu, $\lambda \in \mathbb{R}$ ja $\lambda_1 = \min_{t \in T} g(t, T)$,*

1. *kui $\lambda < \lambda_1$, siis T on λ järgi iseenda vähim alampuu;*
2. *kui $\lambda = \lambda_1$, siis T on λ järgi iseenda optimaalne alampuu, aga mitte vähim;*
3. *kui $\lambda > \lambda_1$, siis T ei ole λ järgi iseenda optimaalne alampuu.*

Tõestus. Olgu T puu, $\lambda \in \mathbb{R}$, $\lambda_1 = \min_{t \in T} g(t, T)$ ja $T_1 = T(\lambda_1)$.

1. Olgu $\lambda < \lambda_1$, siis λ_1 definitsiooni kohaselt kehtib iga $t \in T - \tilde{T}$ korral võrratus

$$\frac{R(t) - R(T_t)}{|\tilde{T}_t| - 1} \geq \lambda_1 > \lambda$$

ehk $R(t) + \lambda > R_\lambda(T_t)$, seega lausest 1.5 tulenevalt $T = T(\lambda)$.

2. Olgu $\lambda = \lambda_1$, siis definitsioonist järeldub $R(t) + \lambda \geq R_\lambda(T_t)$ iga $t \in T - \tilde{T}$, mistõttu lausest 1.4 järeldub seos $T \in \mathcal{T}_\lambda(T)$. Samas aga leidub $t_1 \in T - \tilde{T}$, et $R(t_1) + \lambda = R_\lambda(T_{t_1})$ ehk tipp mida pügades puu kaalutud risk ei vähene, aga saame vähima alampuu $T_1 = T(\lambda)$. Seega $T \neq T_1$ ehk T ei ole iseenda vähim alampuu λ järgi, aga on üks optimaalsetest.

3. Olgu $\lambda > \lambda_1$ ja $T_1 = T(\lambda_1)$, siis vastavalt lause eelmisele osale T on iseenda üks optimaalne alampuu λ_1 järgi ehk $R^*_{\lambda_1}(T) = R_{\lambda_1}(T) = R_{\lambda_1}(T_1)$ ja

$$R(T) + \lambda_1|\tilde{T}| = R(T_1) + \lambda_1|\tilde{T}_1| \Leftrightarrow R(T_1) - R(T) = \lambda_1(|\tilde{T}| - |\tilde{T}_1|).$$

Kuna $\lambda_1 < \lambda$, siis

$$R(T_1) - R(T) < \lambda(|\tilde{T}| - |\tilde{T}_1|).$$

Sellest omakorda järeldub, et $R(T_1) + \lambda|\tilde{T}_1| < R(T) + \lambda|\tilde{T}|$ ehk $R_\lambda(T_1) < R_\lambda(T)$, mistõttu T ei ole λ järgi üks enda optimaalsetest alampuudest.

□

Lause 1.8 (Breiman jt, 1984, Teoreem 10.11)**.** *Olgu T puu, $\lambda_1 = \min_{t \in T - \tilde{T}} g(t, T)$ ja $T_1 = T(\lambda_1)$. Olgu $t \in T_1 - \tilde{T}_1$ ja $T_{1,t}$ tipust t algav puu T_1 haru, siis*

$$g(t, T_1) \begin{cases} > g(t, T), & \textit{kui } T_{1,t} \prec T_t; \\ = g(t, T), & \textit{kui } T_{1,t} = T_t. \end{cases}$$

Tõestus. Kehtigu eeldused, kusjuures $\{t\} \prec T_{1,t} \preceq T_t$. Kui $T_{1,t} = T_t$, siis

$$g(t, T_1) = \frac{R(t) - R(T_{1,t})}{|\tilde{T}_{1,t}| - 1} = \frac{R(t) - R(T_t)}{|\tilde{T}_t| - 1} = g(t, T)$$

Olgu $T_{1,t} \prec T_t$. Esmalt näitame, et $T_{1,t} = T_t(\lambda_1)$ ehk puu T_t vähim alampuu λ_1 järgi on $T_{1,t}$. Järeldusele 1.6 tuginevalt

$$T_1 = \{t' \in T : R(s) + \lambda_1 > R_{\lambda_1}(T_s) \quad \text{iga } t' \text{ eellase } s \text{ korral}\}.$$

Kuna iga puu $T_{1,t}$ leht on ka puu T_1 leht, siis ka

$$T_{1,t} = \{t' \in T_t : R(s) + \lambda_1 > R_{\lambda_1}((T_t)_s) \quad \text{iga } t' \text{ eellase } s \in T_t \text{ korral}\} = T_t(\lambda_1).$$

Lause 1.7 järgi kui $\lambda < \lambda_1$, siis T_t on enda vähim alampuu λ järgi ehk $T_t = T_t(\lambda)$. Seega eeldusest $T_{1,t} = T_t(\lambda_1) \prec T_t(\lambda) = T_t$ ja lausest 1.3 järeldub, et

$$\frac{R(T_{1,t}) - R(T_t)}{|\tilde{T}_t| - |\tilde{T}_{1,t}|} \leq \lambda_1.$$

Teisalt, kui $\lambda > \lambda_1$ on piisavalt suur ehk selline, et

$$R(t) + \lambda \leq R(T_t) + \lambda|T_t| \Leftrightarrow \lambda \geq \frac{R(t) - R(T_t)}{|\tilde{T}_t| - 1},$$

siis $T_t(\lambda) = \{t\} \prec T_t(\lambda_1) = T_{1,t}$ ja lausest 1.3 järeldub, et

$$\lambda_1 < \frac{R(t) - R(T_{1,t})}{|\tilde{T}_{1,t}| - 1}.$$

Kokkuvõttes

$$\frac{R(T_{1,t}) - R(T_t)}{|\tilde{T}_t| - |\tilde{T}_{1,t}|} \leq \lambda_1 < \frac{R(t) - R(T_{1,t})}{|\tilde{T}_{1,t}| - 1}.$$

Seega

$$R(t) - R(T_t) = R(t) - R(T_{1,t}) + R(T_{1,t}) - R(T_t) <$$

$$< (R(t) - R(T_{1,t}))\left(1 + \frac{|\tilde{T}_t| - |\tilde{T}_{1,t}|}{|\tilde{T}_{1,t}| - 1}\right) = (R(t) - R(T_{1,t}))\left(\frac{|\tilde{T}_t| - 1|}{|\tilde{T}_{1,t}| - 1}\right)$$

ehk

$$g(t,T_1)=\frac{(R(t)-R(T_{1,t})}{|\tilde{T}_{1,t}|-1}>\frac{(R(t)-R(T_t)}{|\tilde{T}_t|-1}=g(t,T).$$

□

1.6 Breimani pügamisteoreem

Teoreem 1.9 (Breiman jt, 1984, lk 288-290, Breimani pügamisteoreem)**.** *Olgu puud* $T:=T_0\succ T_1\succ\ldots\succ T_r=\{t_0\}$ *ja* $\lambda_1,\ldots,\lambda_r\in\mathbb{R}$ *leitud vastavalt alapeatükis 1.4 kirjeldatud algoritmile. Siis*

1. $-\infty=:\lambda_0<\lambda_1<\ldots<\lambda_r<\lambda_{r+1}:=\infty$;
2. *kui* $\lambda\in[\lambda_k,\lambda_{k+1}), k\in\{0,1,\ldots,r\}$, *siis* $T(\lambda)=T(\lambda_k)=T_k$.

Tõestus. Kehtigu eeldused ja olgu $r\geq 1$.

1. Kuna λ_1 on lõplik arv, seega kehtib võrratus $-\infty<\lambda_1$ ja sarnaselt ka $\lambda_r<\infty$. Olgu $r>1$ ja $t\in T_1-\tilde{T}_1$. Vastavalt T_1 definitsioonile $g(t,T_0)>\lambda_1$ ehk iga sisetipu, mille korral see võrratus ei kehti, oleme puu T_1 saavutamiseks ära püganud. Lausest 1.8 järeldub, et

   $$g(t,T_1)\geq g(t,T_0)>\lambda_1$$

   ehk $\lambda_1<\min_{t\in T_1-\tilde{T}_1}g(t,T_1)=\lambda_2$. Sarnaselt kehtivad võrratused $\lambda_2<\lambda_3<\ldots<\lambda_r$.

2. Tõestame väite induktsiooniga k järgi. Juhul $k=0$ järeldub lausest 1.7, et iga $\lambda<\lambda_1$ korral $T_0(\lambda)=T_0$. Olgu $r>1$ ja $k=1$ ehk $\lambda\in[\lambda_1,\lambda_2)$. Kuna $\lambda\leq\lambda_2$, siis lause 1.7 järgi kehtib võrdus $T_1(\lambda)=T_1$. Kuna $\lambda\geq\lambda_1$, siis lause 1.3 esimesest väitest järeldub, et

   $$T_0(\lambda)\preceq T_0(\lambda_1)=T_1\prec T_0.$$

Viimane võrratus täidab lause 1.2 eeldused ehk $T_0(\lambda) = T_1(\lambda) = T_1 = T_0(\lambda_1)$.

Olgu $r \geq k$, puu T_{k-1} mittetriviaalne ning kehtigu väide juhuni $k-1$, näitame, et see kehtib ka k korral. Olgu $\lambda \in [\lambda_k, \lambda_{k+1})$. Vastavalt induktsiooni eeldusele kehtib seos $T_1(\lambda) = T_1(\lambda_k) = T_k$. Kuna $\lambda > \lambda_1$, siis lause 1.3 esimese osa järgi kehtib seos $T_0 \succ T_1 = T_0(\lambda_1) \succeq T_0(\lambda)$. Seega lause 1.2 järgi $T_0(\lambda_k) = T_0(\lambda) = T_1(\lambda) = T_k$.

□

2 Klassifikatsiooni- ja regressioonipuud masinõppes

Klassifikatsiooni- ja regressioonipuud on masinõppes kasutusel just klassifikatsiooni- ja regressiooniülesannetes. Järgmised kaks alapeatükki selgitavad, mis need ülesanded on ja kuidas neid CART-puudega lahendada. Peatükk põhineb professor Jüri Lemberi „Tehisõpe I" konspekti peatükkidel 1.1 ja 6.3.5 (Lember, 2021).

2.1 Klassifikatsioon

Klassifikatsioon tähendab objektide liigitamist etteantud klassidesse nende tunnuste alusel. Klassifikatsioon eeldab, et kõik võimalikud klassid on meile ette teada. Kui meil on k klassi, siis nende hulka tähistame $\mathcal{Y} := \{0, 1, \ldots, k-1\}$. Klassifitseeritavatele objektidele seame vastavusse tunnusvektorid $x \in \mathbb{R}^d$, kus d on tunnuste arv. Seega ülesandes on antud hulk $t_0 = \{(x_i, y_i) \mid i = 1, \ldots, n\}$, kus n on objektide arv, iga $i = 1, \ldots, n$ korral $x_i \in \mathbb{R}^d$ ja $y_i \in \mathcal{Y}$. Klassifitseerija on funktsioon, mis seab tunnusvektorile vastavusse ühe klassi:

$$g(x) : \mathbb{R}^d \to \mathcal{Y}.$$

Tipu riskina kasutame funktsiooni $\phi : [0, 1]^k \to \mathbb{R}_0^+$ (*impurity function*), mille argumendid on vastavalt klasside arvule k arvud $p_0(t), \ldots, p_{k-1}(t)$, mis on vastavate klasside osakaalud hulgas t. Funktsioon ϕ peab olema sümmeetriline ehk argumentide järjekorrast sõltumatu, et vältida klasside järestusest tulenevat nihet (*bias*). Tipu t korral võime lühidalt kirjutada ka

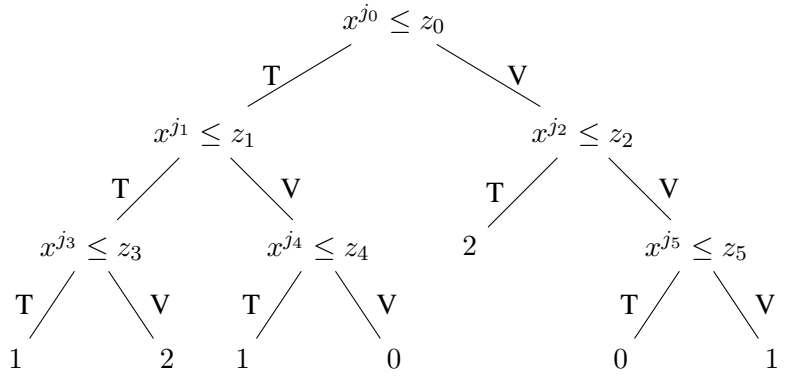
$$\phi(t) := \phi(p_0(t), \ldots, p_{k-1}(t)).$$

Ehk iga tipu t korral on risk $R(t, \phi) := |t|\phi(t)$ ja karistusliige $U(t) = 0$. Enamasti on kasutusel funktsioonid:

- empiiriline risk ehk klassifitseerimisviga: $\phi(p_0, \ldots, p_{k-1}) = 1 - \max_{i=0,\ldots,k-1}(p_i)$,

- Gini indeks: $\phi(p_0, \ldots, p_{k-1}) = 1 - \sum_{i=0}^{k-1} p_i^2$,
- entroopiafunktsioon: $\phi(p_0, \ldots, p_{k-1}) = -\sum_{i=0}^{k-1} p_i \log_2 p_i$.

Karistusliikmest U ei ole klassifikatsioonis kasu, kuna eesmärgiks ei ole tingimata hoida sama klassi objekte koos, vaid objekte klassidesse määrata võimalikult heade reeglite abil. Ühte klassi kuuluvad objektid võivad tunnuste poolest erineda, mistõttu võib olla mitu erinevat tingimuste ahelat, mis määravad objekte samasse klassi. See tähendab, et karistusliikme kasutamine soosib liigset lihtsustamist, mis muudab puu vähem informatiivseks. Seega on klassifikatsioonis mõistlik alati valida iga tipu t korral $U(t) = 0$.



Joonis 3. Näide klassifikatsioonipuust, kus T on tõene, V väär.

Enne puu kasvatamist valime hulgast t_0 juhuslikult kaks hulka - valideerimisandmestiku t_{val} ja testandmestiku t_{test}, millega peale puu kasvatamist seda vastavalt pügame ja pügatud alampuid hindame. Tavaliselt on nendes hulkades $10\% - 20\%$ esialgsetest andmetest. Hulga t_0 alamhulka $t_{train} := t_0 \setminus (t_{val} \cup t_{test})$ nimetame treeningandmestikuks ja sellega kasvatame puud. Valideerimisandmestik aitab pügamisel vähendada ülesobitatust ning testandmestik on oluline objektiivse hinnangu andmiseks puule.

Kuna puude eesmärk on ennustada objekti klassi selle tunnuste järgi ja me teame oma esialgsete andmete klasse, siis soovime leida nii-öelda mehhanismi, mille alusel objektid

hulgast t_{train} jaotuvad oma klassidesse. Seega kõikide võimalike tükelduste asemel tükeldame tippe igal sammul vaid ühe tunnuse väärtuste järgi.

Klassifikatsioonipuud kasvatame rekursiivselt treeningandmetel t_{train} igal sammul valides tunnuse $j \in \{0, \ldots, d-1\}$ ja selle väärtuse $z \in \mathbb{R}$, et leida sel sammul uuritava tipu $t \subseteq t_{train}$ tükeldus

$$t_1 = \{x \in t : x^j \leq z\}, \quad t_2 = \{x \in t : x^j > z\},$$

mis minimeeriks summat

$$|t_1|\phi(p_0(t_1), \ldots, p_{k-1}(t_1)) + |t_2|\phi(p_0(t_2), \ldots, p_{k-1}(t_2)) = R(t_1) + R(t_2).$$

Kui tipu kõik objektid kuuluvad samasse klassi ehk $\phi(t) = 0$ või tükeldus ei vähenda riski ehk $R(t_1) + R(t_2) < R(t)$, siis me seda tippu edasi ei tükelda. Kui kasvatame tipule lapsed, siis fikseerime sellele tipule tunnuse j ja vastava väärtuse z, millede järgi seda tippu tükeldasime. Igale tipule määrame vastavusse klassi, mis on selle objektide kõige sagedam klass. Lõpuni kasvatatud puud tähistame T_0.

Puu T_0 pügamiseks kasutame valideerimisandmestiku t_{val}. Ehk täidame puu T_0 objektidega hulgast t_{val}, järgides tippudel fikseeritud tükeldamiste reegleid. Koheselt pügame kõik tipud, mille vähemalt ühte lapsesse ei jõudnud ükski hulga t_{val} objekt. Pügamisel võime kasutada erinevat ϕ, kui kasvatamisel ning praktikas tihti ka nii tehakse. Näiteks kui kasvatamisel kasutasime Gini indeksit, siis võime pügamiseks valida entroopiafunktsiooni ϕ rolli.

Iga puu $T \preceq T_0$ risk on defineeritud

$$R(T, \phi) = \sum_{t \in \tilde{T}} R(t, \phi) = \sum_{t \in \tilde{T}} |t|\phi(t)$$

ning reaalarvu λ järgi kaalutud risk on

$$R_\lambda(T, \phi) = \sum_{t \in \tilde{T}} |t|\phi(t) + \lambda|\tilde{T}|.$$

Pügame puud T_0 vastavalt alapeatükis 1.4 kirjeldatud algoritmile ja saame sarnaselt vähimate alampuude jada ja nendele vastavate reaalarvude λ_i jada.

Peale pügamist hindame iga vähima alampuu täpsust testandmestikul t_{test}. Puu täpsuseks nimetame osakaalu t_{test} objektidest, mille puu määrab õigesse klassi. Klassifitseerijaks g valime vähimatest alampuudest kõige kõrgema täpsusega puu.

Tunnused võivad olla ka kvalitatiivsed (näiteks värv või asukoht), mille puhul on kaks lähenemist. Esimene lähenemine on anda neile numbrilised väärtused (näiteks roheline olgu 0, sinine 1, punane 2, $\dots$), kuid nii võivad tekkida ekslikud seosed tulenevalt juhuslikult määratud numbrilisest järjestusest. Teine lähenemine on kasutada *one-hot encoding* meetodit ehk luua igale kvalitatiivse tunnuse väärtusele vastav binaarne tunnus (kui esialgsel tunnusel on antud väärtus, siis 1, muidu 0). *One-hot encoding* meetodi kasutamine on järjestusagnostiline ehk järjestusest sõltumatu ning seega tihti eelistatum.

Klassifikatsioon on kasutusel masinõppes, kus soovime vastuseks mingit kindlat klassi. Kahe klassiga probleemide vastuseks on enamasti jah/ei (tõene/väär) ning sellisteks probleemideks on näiteks, kas meil on rämpspost, kas patsiendil on antud haigus või kas Titanicu reisija jäi ellu. Rohkemate klassidega ülesandeks on näiteks diagnoosida patsiendile haigus.

2.2 Regressioon

Regressiooni korral on meil iga treening-, valideerimis- ja testandmestiku objekti korral teada sellele vastav reaalarvuline väärtus, mitte klass. Ehk esialgseks objektide hulgaks

on

$$t_0 = \{(x_i, y_i) | i = 1, \ldots, n\},$$

kus n on objektide arv, iga $x_i \in \mathbb{R}^d$ ja iga $y_i \in \mathbb{R}$. ning regressioonifunktsioon on funktsioon

$$g(x) : \mathbb{R}^d \to \mathbb{R}.$$

Sarnaselt klassifikatsioonile jaotame oma esialgse hulga t_0 kolmeks hulgaks t_{train}, t_{val} ja t_{test}.

Regressiooni eesmärk on ennustada igale objektile reaalarvuline väärtus, mistõttu täpsuse asemel kasutatakse vähimruutude kriteeriumit ehk ennustuse ja tegeliku väärtuse erinevuse ruutu. Tipu t_i ennustavaks väärtuseks on sellesse sattunud treeningandmetele vastavusse seatud reaalarvude aritmeetiline keskmine ehk $\widehat{c}_i = \frac{1}{|t_i|} \sum_{x_j \in t} y_j$. Iga tipu t risk on seega $R(t) = \sum_{x_i \in t} (y_i - \widehat{c}_i)^2$. Sarnaselt klassifikatsioonile on mõistlik iga sisetipu karistusliige võrdsustada nulliga, et hoiduda liigsetest lihtsustustest.

Sarnaselt klassifikatsioonile valime puu kasvatamise igal sammul sellise tunnuse ja selle reaalarvulise väärtuse, mille alusel tükeldame hulka, et vähimruutude summa oleks minimaalne. Puud kasvatame sarnaselt klassifikatsioonile ehk rekursiivselt igal kasvatamise sammul leiame sellised $j \in \{1, \ldots, d\}$ ja $z \in \mathbb{R}$, et käesoleva hulga $t \subseteq t_{train}$ tükeldus

$$t_1 = \{x \in t : x^j \leq z\}, \quad t_2 = \{x \in t : x^j > z\},$$

minimeeriks vähimruutude summat

$$\min_{c_1} \sum_{x_i \in t_1} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in t_2} (y_i - c_2)^2 = R(t_1) + R(t_2)$$

ja kehtiks

$$R(t_1) + R(t_2) < R(t).$$

Sarnaselt klassifikatsioonile fikseerime tükeldavale tipule vastavusse tunnuse j, selle väärtuse z ning tükeldustele nende ennustatavad väärtused. Lõpuni kasvatatud puud tähistame T_0.

Iga puu $T \preceq T_0$ riskiks on

$$R(T) = \sum_{t \in \tilde{T}} R(t) = \sum_{t \in \tilde{T}} \sum_{x_i \in t} (y_i - \widehat{c})^2$$

ja kaalutud risk $\lambda \in \mathbb{R}$ järgi $R_\lambda(T) = R(T) + \lambda|\tilde{T}|$. Sarnaselt klassifikatsioonile püga-me puu alapeatükis 1.4 kirjeldatud algoritmi järgi, kasutades valideerimisandmestikku t_{val}. Saame vähimate alampuude ja reaalarvude jada, millest igale alampuule arvutame testandmestikul t_{test} riski. Regressioonifunktsiooniks valime T_0 vähima alampuu, mille ennustused testandmestikul t_{test} on tegelikkusele kõige lähedasemad ehk minimaalse riskiga.

Regressiooni kasutatakse, kui eesmärgiks on ennustada arvulist väärtust objektile, näiteks kinnisvara turuväärtust või aktsia õiglast hinda.

3 Klassifitseerimisnäide

Selles peatükis lahendatakse üks populaarseim klassifikatsiooni ülesanne masinõppe võistluskeskkonnas Kaggle - „Titanic – Machine Learning from Disaster" - klassifikatsioonipuuga. Ülesande eesmärgiks on leida klassifitseerija, millega ennustada võimalikult täpselt, kas Titanicu reisija jäi ellu (Cukierski, 2012). Peatükis selgitatava lahenduse programm on avalikult kättesaadav Kaggle'i keskkonnas (Jesse, 2025b).

Näidise jaoks kasutame reisija tunnustena pileti numbrilist klassi (*Pclass*), sugu (*Sex*), vanust (*Age*), õdede-vendade ja abikaasade arvu (*SibSp*), vanemate ja laste arvu (*Parch*), piletihinda (*Fare*), pardalemineku sadamat (*Embarked*). Tulemuseks on, kas reisija jäi ellu või mitte (*Survived*), vastavalt andmetes 1 või 0. Ainsad kaks kvalitatiivset tunnust on reisija sugu ja pardalemineku sadam. Soo saame binaarselt jaotada meestele andes väärtuse 0 ja naistele väärtuse 1. Sadamaid oli kolm - Cherbourg (*C*), Queenstown (*Q*) ja Southampton (*S*). Numbrite nullist kaheni vastavusse seadmine tähendaks ekslikku õpitavat järestust sadamate vahel, mida me ei soovi, seega kasutame *one-hot encoding* meetodit ehk igale sadamale loome enda tulba, kus on vastavalt 1 või 0, vastavalt, kas reisija tuli pardale antud sadamas või mitte. Andmed on tabelis 1 kujutatud viisil. Indekseerime tunnused vasakult paremale nagu tabelis 1 ehk $\mathit{Pclass} \to 0, \mathit{Sex} \to 1, \ldots, S \to 8$.

Tabel 1. Titanicu reisijate andmed.

$Survived$	$Pclass$	Sex	Age	$SibSp$	$Parch$	$Fare$	C	Q	S
0	3	0	22	1	0	7,2500	0	0	1
1	1	1	38	1	0	71,2833	1	0	0
1	3	1	26	0	0	7,9250	0	0	1
1	1	1	35	1	0	53,1000	0	0	1
0	3	0	35	0	0	8,0500	0	0	1
...	...	...	...	...	...	...	...	...	...

Toimime edasi vastavalt klassifikatsiooni käsitlevale alapeatükile 2.1. Esmalt jagame algsed andmed treeningandmeteks t_{train}, valideerimisandmestikuks t_{val} ja testandmestikuks t_{test}. Valime funktsiooniks ϕ Gini indeksi ning kuna meil on võimalik ennustada kahte

klassi 0 ehk reisija uppus või 1 ehk jäi ellu, siis

$$\phi(p) = 2p(1-p),$$

kus p võib olla nii ellujääjate kui ka uppunute osakaal. Tipu t riski defineerime

$$R(t,\phi) = |t|\phi\left(\frac{|\{x_i \in t | y_i = 1\}|}{|t|}\right).$$

Kasvatame puud treeningandmestikul t_{train} rekursiivselt, valides igal sammul tunnuse indeksiga j ja selle väärtuse z, mis minimeerivad tipu tükelduse riski. Igale kasvatatud tipule fikseerime ka ennustuse ehk kas enim sinna jõudnud reisijaid treeningandmestikus uppus (0) või jäid ellu (1). Tipu tükeldamise lõpetame ära ka juhul, kui kõik sinna jõudnud reisijad jäid ellu või kõik surid, või minimaalse riskiga tükelduse risk on suurem kui tipu risk. Kasvatamise lõpuks jõuame puuni T_0.

Lõpuni kasvatatud puu täpsuses testandmestikul on $0{,}7123$, kuid treeningandmetel $0{,}9883$ ehk puu on ülesobitatud. Täpsuse suurendamiseks pügame puud, kasutades selleks valideerimisandmestikku t_{val} ja ϕ jaoks klassifitseerimisviga ehk

$$\phi(p) = \min(1-p, p).$$

Koheselt pügame ära kõik tipud, mille lastest vähemalt ühte ei jõudnud üksi valideerimisandmetes olev reisija. Puud pügades saame vähimate alampuude jada ja arvutame igale neist täpsuse nagu tabelis 2.

Tabelist 2 näeme, et λ_1 on negatiivne ehk puust T_0 puu T_1 saamiseks pügatud tippudest algavad harud suurendasid puu riski. Samas ka, et $\lambda_2 = 0$, siis pügati vaid tippe puust T_1, millel oli võrdne risk endast algava haru riskiga.

Kõige kõrgema täpsusega on puu T_6 ning seega valime selle oma klassifitseerijaks g. Puu täpsus võistlusandmestikul on $0{,}780$, mis on parem, kui kasutada nii kasvatamiseks kui

Tabel 2. Masinõppe näite pügatud alampuud ja nende täpsused.

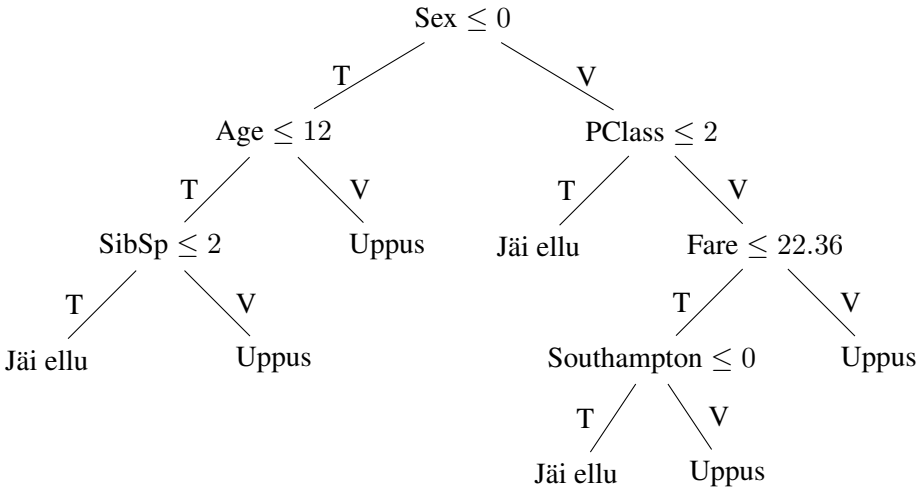
T_i	λ_i	R	$\lvert\tilde{T}_i\rvert$	Täpsus
T_0	$-\infty$	22	32	0,765
T_1	$-2{,}22 * 10^{-16}$	22	30	0,811
T_2	0	22	25	0,811
T_3	$1{,}33 * 10^{-15}$	22	24	0,811
T_4	0,33	23	21	0,814
T_5	0,4	25	16	0,825
T_6	0,89	33	7	0,828
T_7	1,0	36	4	0,793
T_8	1,5	39	2	0,782
T_9	32,5	73	1	0,67

ka pügamiseks Gini indeksit ($0{,}751$) ja marginaalselt parem, kui kasutada klassifitseerimisviga ($0{,}778$).

Võistlusandmestik on andmestik, mis ei ole võistlejale kättesaadav ja sellega määratakse klassifitseerija - antud juhul puu - lõplik täpsus.

Puu T_6 on kujutatud joonisel 4. Jooniselt näeme, et kõige esimene ja seega kõige olulisem tunnus on inimese sugu - meestest enamus uppusid ja naised jäid ellu. Meessoost reisijatele ennustab puu ellujäämist vaid kuni 12-aastastel, kellel oli pardal kaasas kuni kaks õde-venda. Naissoost reisijate ellujäämise ennustust mõjutavad pileti klass, hind ja pardalemineku sadam.

Täpsema puu treenimiseks oleks võimalik muuta treening- ja valideerimisandmete osakaalu, proovida erinevaid ϕ-sid ning kasutada rohkem tunnuseid reisijate kohta.



Joonis 4. Titanicu ülesande puu T_6, kus T on tõene, V väär.

4 Klasterdusülesande näide

Selles peatükis lahendatakse probleem, mis motiveeris antud bakalaureusetöö kirjutamist ning tekitas vajaduse karistusliikme kasutuselevõtu järele. Probleemiks on klasterdada n elementi, mille kohta on defineeritud iga kahe elemendi vaheline sümmeetriline mõõdik vahemikus $[0, 1]$. Kahe elemendi vaheline mõõdik 1 tähendab, et need kaks elementi peaksid olema kindlasti ühes klastris, ja 0, et kindlasti mitte. Sellest tekkis ka vajadus karistusliikme järgi, kuna ilma selleta võib kasvatamise algoritm viia läbi tükeldusi, mis lahutavad kaks elementi, mis peaksid sattuma ja jääma samasse klastrisse. Peatükk on loodud toetudes autori loodud programmile, mis on avalikult kättesaadav GitHubi keskkonnas (Jesse, 2025a).

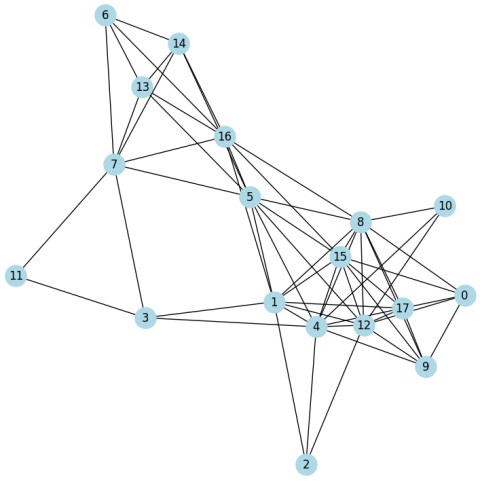
Probleem kerkis bioinformaatikas esineva geneetilise kolokalisatsiooni tulemuste analüüsimisel. Kolokalisatsiooni eesmärk on tuvastada, kas kahel geneetilisel tunnusel, näiteks indiviidi pikkusel ja teatud geeni ekspressioonil, on mingis geenipiirkonnas ühine põhjuslik variant. Geneetiline variant (kindel genoomi positsioon ja sellel esinev alleel) on kahe tunnuse jaoks põhjuslik, kui selle alleel mõjutab oluliselt mõlemat tunnust. Mõõdik lähedal 1-le kahe tunnuse vahel viitab jagatud põhjuslikule variandile ehk tunnuste kolokalisatsioonile, mõõdik lähedal 0-le, et tunnused on sõltumatud (Giambartolomei jt, 2014). Tihti tekivad kõrgete omavaheliste mõõdikutega tunnuste ahelad, mille esimese ja viimase tunnuse mõõdik on lähedal nullile. Oluline on pidada meeles, et puuduvad empiirilised andmed, mille pealt klasterdusi hinnata.

Olgu meil nüüd 18×18 sümmeetriline maatriks Y, nagu tabelis 3, mille i-nda rea ja j-nda veeru element on $y_{ij} \in [0, 1]$.

Kujutades maatriksit Y graafina ja joonistades välja vaid need servad, mille mõõdiku väärtus on vähemalt $0{,}8$, saame tulemuseks graafi joonisel 5. Lävend $0{,}8$ on juhuslik ning servade pikkused graafil ei oma tähendust, graaf on mõeldud eelkõige intuitsiooni loomiseks. Parimaks intuitiivseks klastrite indikaatoriks on alamgraafide tihedused

Tabel 3. Y maatriks.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	1,00	0,78	0,65	0,64	0,77	0,74	0,07	0,11	0,85	0,97	0,65	0,01	0,91	0,47	0,15	0,92	0,77	0,99
1	0,78	1,00	0,83	0,83	0,98	0,92	0,43	0,56	0,92	0,77	0,76	0,45	0,97	0,73	0,69	0,94	0,81	0,95
2	0,65	0,83	1,00	0,52	0,83	0,71	0,36	0,45	0,76	0,59	0,48	0,39	0,80	0,52	0,56	0,76	0,63	0,76
3	0,64	0,83	0,52	1,00	0,81	0,79	0,63	0,81	0,78	0,54	0,48	0,83	0,79	0,52	0,67	0,76	0,73	0,71
4	0,77	0,98	0,83	0,81	1,00	0,91	0,11	0,14	0,93	0,83	0,81	0,05	0,97	0,64	0,32	0,95	0,77	0,95
5	0,74	0,92	0,71	0,79	0,91	1,00	0,73	0,81	0,93	0,65	0,63	0,59	0,83	0,88	0,91	0,90	0,90	0,74
6	0,07	0,43	0,36	0,63	0,11	0,73	1,00	0,83	0,06	0,07	0,28	0,79	0,30	0,82	0,88	0,06	0,83	0,04
7	0,11	0,56	0,45	0,81	0,14	0,81	0,83	1,00	0,06	0,12	0,41	0,90	0,45	0,86	0,92	0,07	0,88	0,06
8	0,85	0,92	0,76	0,78	0,93	0,93	0,06	0,06	1,00	0,86	0,82	0,01	0,93	0,55	0,15	0,95	0,87	0,83
9	0,97	0,77	0,59	0,54	0,83	0,65	0,07	0,12	0,86	1,00	0,70	0,08	0,87	0,42	0,15	0,88	0,69	0,95
10	0,65	0,76	0,48	0,48	0,81	0,63	0,28	0,41	0,82	0,70	1,00	0,34	0,80	0,55	0,53	0,79	0,66	0,74
11	0,01	0,45	0,39	0,83	0,05	0,59	0,79	0,90	0,01	0,08	0,34	1,00	0,44	0,17	0,52	0,01	0,68	0,00
12	0,91	0,97	0,80	0,79	0,97	0,83	0,30	0,45	0,93	0,87	0,80	0,44	1,00	0,64	0,52	0,94	0,79	0,96
13	0,47	0,73	0,52	0,52	0,64	0,88	0,82	0,86	0,55	0,42	0,55	0,17	0,64	1,00	0,96	0,55	0,92	0,54
14	0,15	0,69	0,56	0,67	0,32	0,91	0,88	0,92	0,15	0,15	0,53	0,52	0,52	0,96	1,00	0,19	0,96	0,12
15	0,92	0,94	0,76	0,76	0,95	0,90	0,06	0,07	0,95	0,88	0,79	0,01	0,94	0,55	0,19	1,00	0,82	0,93
16	0,77	0,81	0,63	0,73	0,77	0,90	0,83	0,88	0,87	0,69	0,66	0,68	0,79	0,92	0,96	0,82	1,00	0,72
17	0,99	0,95	0,76	0,71	0,95	0,74	0,04	0,06	0,83	0,95	0,74	0,00	0,96	0,54	0,12	0,93	0,72	1,00



Joonis 5. Graaf nummerdatud tippudega.

ehk mis osakaalul alamgraafi tippudel on üksteisega servad. Graafilt 5 näeme, et tipud $6, 7, 13, 14, 16$ võiksid sattuda ühte klastrisse ning tipud $0, 1, 2, 4, 5, 8, 9, 10, 12, 17$ teise. Edaspidi analüüsime klasterdust CART-puudega ja demonstreerime vajadust karistusliikme kasutuselevõtuks.

Defineerime kahe funktsiooni eeskirjad

$$\phi_0 : [0,1] \to \mathbb{R}^+ \text{ ja } \phi_1 : [0,1] \to \mathbb{R}^+.$$

Funktsioonide ϕ_0 ja ϕ_1 edasised valikud on üksnes näite illustreerimiseks. Nende funktsioonide abil defineerime lehe riski

$$R(t) := \sum_{(i,j)\in t\times t} \phi_1(y_{ij}),$$

karistusliikme

$$U(\{t_1, t_2\}) := \sum_{(i,j)\in t_1\times t_2} \phi_0(y_{ij})$$

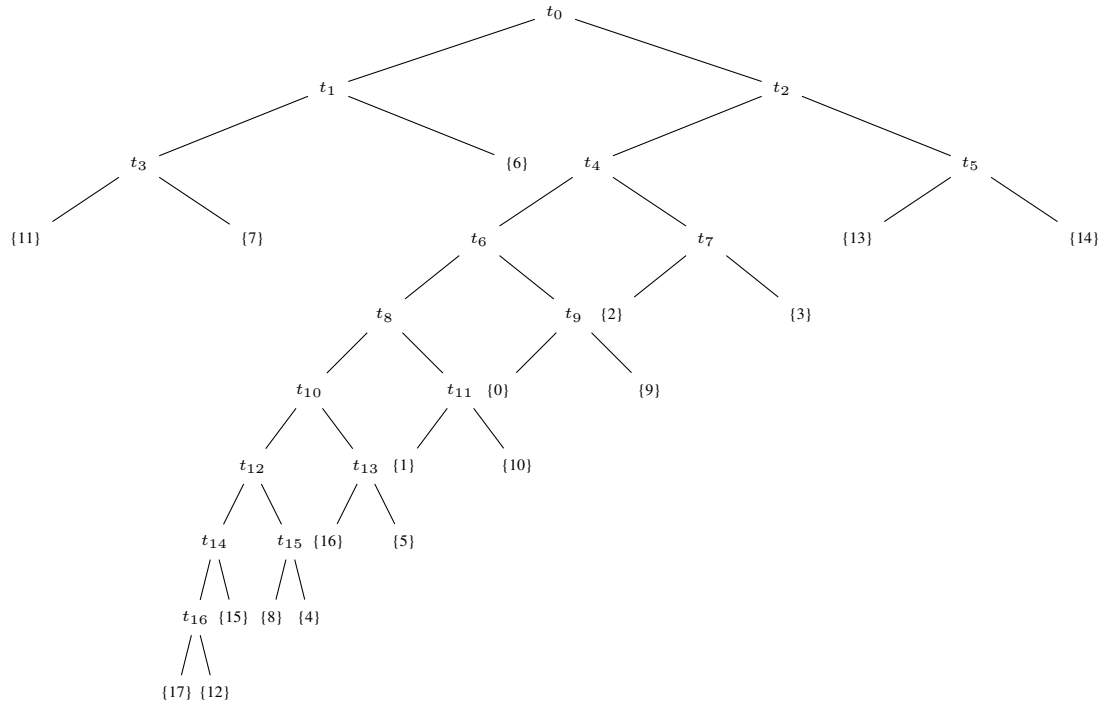
ja puu riski

$$R(T) := \sum_{t\in\tilde{T}} \sum_{(i,j)\in t\times t} R(t) + \sum_{t\in T-\tilde{T}} U(t).$$

Vaatame, mis juhtub, kui $\phi_0(y) = 0$, ehk karistusliige puudub ($U(t) = 0$). Valime $\phi_1(y) = \sqrt{1-y}$ ning kasvatame juurest $t_0 = \{0, 1, \ldots, 17\}$ puu T_0 vastavalt alapeatükis 1.2 kirjeldatud algoritmile ehk vaadates igal sammul kõiki võimalikke tükeldusi. Jõuame puuni T_0, mis on kujutatud joonisel 6. Elementide $6, 7, 13$ ja 14 omavahelised mõõdikud on väga kõrged, aga kasvamise algoritm määrab need elemendid erinevatesse peaharudesse. Seetõttu tekib olukord, kus ainuke viis neid tippe ühte klastrisse koondada on kui ainus tipp on juur, mis ei ole rahuldav lahendus. Olenemata riski valikust (antud näites ϕ_1), ilma karistusliikmeta huvitub algoritm ainult võimalikult erinevate elementide paigutamist eri tükkidesse, pööramata tähelepanu sarnaste elementide kooshoidmisele. Seega ei ole võimalik klasterdusülesandes vaid riskile toetudes CART-puid kasutada. Edasi näitame, kuidas karistusliikme kasutuselevõtt selle probleemi lahendab.

Teisalt, kui kasutame karistusliiget ehk valime funktsioonid $\phi_0(y) = \sqrt{y}$ ja $\phi_1(y) = \sqrt{1-y}$ ning kasvatame puu vastavalt alapeatükis 1.2 kirjeldatud algoritmile. Saame puu T_0', mis on kujutatud joonisel 7. Puus T_0' on klastrid intuitiivsemad kui puus T_0.

Pügame puud T_0' vastavalt alapeatükis 1.4 kirjeldatud algoritmile, kasutades samu funktsioone. Tulemusena saame vähimate alampuude jada, $T_1', \ldots, T_5'$, mis tekib pügades



Joonis 6. Puu T_0 ilma karistusliikmeta.

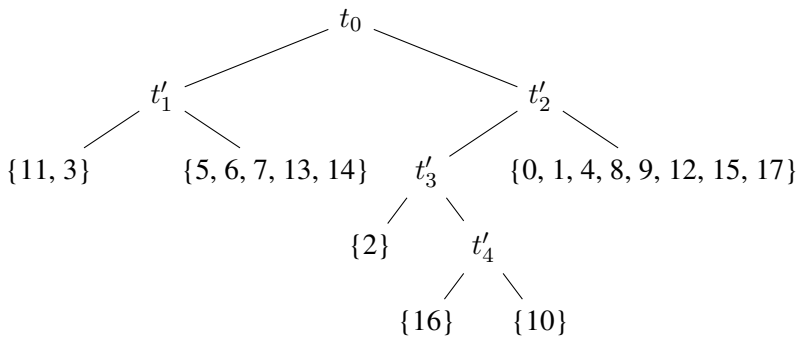
vastavalt tippe t'_4, t'_3, t'_2, t'_1 ja lõpuks juurt t_0. Neile vastavad reaalarvud on

$$\lambda'_1 \approx 0{,}35, \quad \lambda'_2 \approx 1{,}17, \quad \lambda'_3 \approx 2{,}27, \quad \lambda'_4 \approx 3{,}72, \quad \lambda'_5 \approx 59{,}17.$$

Kui aga pügame puud T'_0 funktsioonidega $\phi_0(y) = y$ ja $\phi_1(y) = 1 - y$, saame vähimate alampuude jada T''_1, T''_2, T''_3, pügades puust T'_0 vastavalt tipud t'_2, t'_1 ning t_0. Nendele vastavad:

$$\lambda''_1 \approx -2{,}06, \quad \lambda''_2 \approx 0{,}83, \quad \lambda''_3 \approx 46{,}9.$$

Negatiivne λ''_1 väärtus tähendab, et uued funktsioonid olid konservatiivsemad kui kasvatamisel kasutatud ehk hoiduvad veelgi tükeldamisest. Funktsioonide valik võimaldab puu kasvatamisel ja pügamisel soosida või pärssida tükeldamist. Leitud alampuude seast võime valida endale lõpliku klasterduse. Oleme seega näidanud, et karistusliikme



Joonis 7. Puu T'_0.

kasutamine võimaldab CART-puid kasutada ka klasterdusülesannetes.

Kokkuvõte

Bakalaureusetöö esimeses peatükis üldistasime CART-puude mõistestikku uudse karistusliikmega. Selle raames üldistasime ja vastavalt tõestasime Breimani pügamisteoreemi ning teoreemi kehtivuseks vajalikud laused. CART-puude teooria arendusena on võimalik puude jaoks defineeritud riski edasi üldistada.

Teises peatükis käsitlesime CART-puude kasutust klassifikatsiooni- ja regressiooniülesannetes. Neis ülesannetes on kaks küsimust. Esiteks, kas objektid jaotuvad üldse seaduspäraselt, ning teiseks, milliste seaduspärade järgi see toimub. Karistusliikme kaasamisel jääks esimene küsimus vastamata, mis omakorda võiks viia ekslike järelduste või liigsete lihtsustusteni. Edasine uurimistöö saaks keskenduda sellele, kas ja kuidas on karistusliiget võimalik rakendada klassifikatsiooni- ja regressiooniülesannetes. Kui see osutub võimalikuks, avaneb võimalus uurida karistusliikme kasutamist ka erinevates CART-puudele tuginevates masinõppemeetodites, näiteks juhusliku metsa (*random forest*) mudelites.

Kolmandas peatükis lahendasime klassikalise Titanicu andmestikul põhineva ellujääjate klassifitseerimisülesande. Selle loogiliseks jätkuks oleks katsetada karistusliikme kasutuselevõttu mudelis.

Neljandas peatükis näitasime, et karistusliikme lisamine võimaldab CART-puid rakendada ka klasterdusülesannete lahendamisel. Edasine uurimine saaks keskenduda CART-puude skaleeritavusele klasterdusülesannetes ning võrdlusele alternatiivsete klasterdusmeetoditega.

Kasutatud allikad

Breiman, L., J. H. Friedman, C. J. Stone ja R. A. Olshen (1984). *Classification and Regression Trees*. Taylor & Francis, lk. 279–289. ISBN: 9780412048418. URL: https://books.google.ee/books?id=JwQx-WOmSyQC.

Cukierski, W. (2012). *Titanic - Machine Learning from Disaster*. Kaggle. URL: https://kaggle.com/competitions/titanic (vaadatud 12.05.2025).

Giambartolomei, C., D. Vukcevic, E. E. Schadt, L. Franke, A. D. Hingorani, C. Wallace ja V. Plagnol (2014). "Bayesian test for colocalisation between pairs of genetic association studies using summary statistics". *PLoS genetics* 10.5, e1004383.

Jesse, M. (2025a). *Klasterdusülesande programm*. GitHub. URL: https://github.com/mjesse-github/cart-clustering (vaadatud 21.05.2025).

Jesse, M. (2025b). *Titanicu ellujääjate klassifikatsiooniülesande lahendus CART-puudega*. Kaggle. URL: https://www.kaggle.com/code/mihkeljesse/cart-lahendus (vaadatud 12.05.2025).

Laan, V. (2020). *Diskreetne matemaatika I*. Loengukonspekt. Tartu Ülikool. URL: https://courses.ms.ut.ee/LTMS.00.019/2020_spring/uploads/Main/kon.pdf.

Lember, J. (2021). *TEHISÕPE I*. Loengukonspekt. Tartu Ülikool. URL: https://courses.ms.ut.ee/MTMS.02.046/2021_spring/uploads/Main/tehisÃµpe21.pdf.

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Mihkel Jesse,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose „Breimani pügamisteoreemi üldistus", mille juhendajad on professor Jüri Lember, kaasprofessor Kaur Alasoo ja teadur Ago-Erik Riet, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commonsi litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.

3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.

4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Mihkel Jesse

27.05.2025