# 11. tétel

Berekméri Evelin

2019. június 18.

**Kivonat**

Számítógépes tanulás – Predikciós és klasszifikációs módszerek. Felügyelt és felügyelet nélküli tanítás. A tanítóhalmaz, a validáció és a túlfittelés. K-means, Support Vector Machine, Random Forest, k-NN-módszer.

# 1. Számítógépes tanulás

A gépi tanulás a mesterséges intelligencia egy ága. Olyan módszereket foglal magába, amelyek meglévő adatokból építenek matematikai modelleket ahhoz, hogy szabályszerűségeket ismerjenek fel vagy predikciókat hajtsanak végre - ismeretlen adatokra is. A gépi tanulásos algoritmusokat két nagy csoportra oszthatjuk: felügyelt (supervised) és felügyelet nélküli tanításra (unsupervised learning). Felügyelt tanítás esetén a statisztikai modell "input" adatok alapján prediktál "output" adatokat. A felügyelet nélküli tanítás tárgykörébe tartozó módszerek esetén nem beszélhetünk output adatokról, ugyanis az algoritmus a rendelkezésre álló adathalmaz szerkezetéről és kapcsolatairól szolgál többletinformációval.

# 2. Felügyelt tanítás

A felügyelt tanítás esetén használt adathalmazban minden megfigyelés esetén az input éték(ek)hez tartozik egy output érték. Az input változókat általában X-szel jelöljük ($X = (X_1, X_2, ..., X_p)$ p db változó esetén) és többféleképpen szoktak rá hivatkozni: (független) változók - (independent) variables -, "features", "predictors". Az ouput változókat - másnéven függő változókat (dependent variables) vagy válaszokat - általában Y-nal jelöljük. Azt feltételezzük, hogy az X és az Y között van valamilyen kapcsolat, amelyet a következőképpen írhatunk fel:

$$Y = f(X) + \epsilon,$$

ahol $f$ X ismeretlen függvénye és $\epsilon$ a hiba tag, amely független X-től és az átlaga nulla. $f$ becslése predikció vagy inferencia végrehajtásához szükséges.
A gépi tanulás és a statisztikai tanulás fogalma gyakran összemosódik az emberek fejében, viszont bizonyos források szerint a lényeges különbség a két terület között az a céljük: a gépi tanulás helyes predicióra fókuszál, ezzel ellentétben a statisztikai tanulás célja az X és Y közötti kapcsolat felderítése (inferencia).

## 2.1. Predikció

Sok esetben az X inputhalmaz könnyen elérhető, de az Y nem könnyen hozzáférhető. Ilyenkor, mivel a hibatag nullára átlagolódik, megjósolhatjuk Y-t:

$$\hat{Y} = \hat{f}(X),$$

ahol $\hat{f}$ f becslése és $\hat{Y}$ Y becsült értéke. $\hat{f}$-et ilyenkor többnyire fekete dobozként kezeljük, mivel általában nem ismerjük annak egzakt formáját. $\hat{Y}$ pontossága két tényezőtől függ: a reducibilis és az irreducibilis hibától. Általában $\hat{f}$ nem becsli elég jól $f$-et, ezért ebből is származik egy hiba. Viszont ez a hiba reducibilis, mivel $\hat{f}$ pontosságát tudjuk növelni pontosabb módszerekkel. Ugyanakkor, ha teljes pontosan meg is tudnánk határozni f-et, a megbecsült válasznak $\hat{Y} = f(X)$ formája lenne és ez még így is tartalmazna hibát, mivel Y $\epsilon$ függvénye is,

amit viszont nem lehet X-ből meghatározni. Tehát $\epsilon$ variabilitása is hatással van a predikció pontosságára. Ezt irreducibilis hibának nevezik, mivel akármennyire is pontosan becsüljük meg f-et, nem tudjuk csökkenteni az $\epsilon$ miatt megjelenő hibát. $\epsilon$ magába foglalhat meg nem mérhető tényezőket, illetve olyan meg nem mért változókat, amelyek hasznosak lehetnek Y predikciójában. Ilyenkor Y megbecsült és valódi értéke közötti különbség négyzetének várható értéke a következőképpen áll elő:

$$E[Y - \hat{Y})^2 = E(f(X) + \epsilon - f(\hat{X})]^2 = [f(X) + f(\hat{X})]^2 + Var(\epsilon),$$

ahol $Var(\epsilon)$ $\epsilon$ szórása. Az első tag reducibilis és a második tag az irreducibilis hibát jelöli. Az irreducibilis hiba egy felső határt jelöl ki Y becslésének pontosságára.

## 2.2. Inferencia

Inferencia esetén f egzakt formáját keressük, mivel a célunk nem feltétlenül az, hogy prediktáljunk vele, hanem az, hogy jobban megértsük a kapcsolatot X és Y között (Y hogyan változik X függvényében). Olyan kérdésekre keressük ilyenkor a válasz, mint például:

- mely input változók vannak ténylegesen kapcsolatban az output értékkel?
- milyen kapcsolatban állnak az input változók az output értékkel? mely változók növekedésének hatására növekszik az output, melyekre csökken?
- milyen formában írható fel a kapcsolat az input és az output között? pl. lineáris?

## 2.3. f becslése

Különböző lineáris és nem lineáris módszerek léteznek f megbecslésére. A rendelkezésünkre álló adathalmaznak azt a részét, amelyen betanítjuk a modellünket f megbecslésére, tanító (training) adathalmaznak nevezzük. Ha van n db megfigyelésünk és p db független változónk, ugyanakkor $x_{ij}$ jelöli az x-edik "sorban" levő megfigyeléshez tartozó, j-edik "oszlopban" levő független változót (i = 1, 2, ..., n és j = 1, 2, ..., p), illetve $y_i$ jelöli az i-edik megfigyeléshez tartozó outputot, akkor a training adathalmazunkat a következőképpen írhatjuk fel:

$$\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\},$$

ahol

$$x_i = (x_{i1}, x_{i2}, ..., x_{ip})^T.$$

A cél az, hogy olyan f függvényt találjunk, ami Y-t a lehető legjobban megközelíti, azaz a reducibilis hibát a lehető legjobban csökkenti ($\hat{Y}\hat{f}(X)$. Ehhez különböző paraméteres és nem paraméteres módszerek léteznek. A paraméteres módszerek esetén rögzítjük, hogy f-nek milyen formát feltételezünk (pl. lineárisan függ a független változóktól: $f(X) = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p$) és valamilyen módszerrel (pl. legkisebb négyzetek módszerével) fitteljük a függvényt a megfigyelési adatokra, amivel megkapjuk a $\beta_i$ együtthatókat. Ez magában hordozza annak a kockázatát, hogy f valójában nagy mértékben különbözik attól a formától, amit mi feltételeztünk. Ehhez képest a nem paraméteres módszerek esetén nem teszünk explicit feltételezéseket f formájáról, nem áll fenn a veszélye, hogy a rosszul feltételezett függvény forma miatt nem lesz elég pontos $\hat{f}$, viszont ugyanakkor sokkal több megfigyelésre van szükségünk, mint a paraméteres módszereknél. Ha a célunk az inferencia, ajánlott a kevesebb változót számításba módszereket

választani, mivel ezeket könnyebb értelmezni, ezek viszont általában kevésbé flexibilisek a nemparaméteres módszerekhez képest, amelyeket érdemes inkább akkor választani, ha prediktálni szeretnénk a modellel.
Egy további szempont szerint az adathalmazt két csoportba osztjuk aszerint, hogy kvantitatív vagy kvalitatív adatokat tartalmaznak. A kvantitatív változók numerikus értékeket vesznek fel. A kvalitatív változók K db különböző osztály (kategória) közül vesznek fel egy értéket. Abban az esetben, ha a függő változóink kvantitatívak, regressziót alkalmazunk. Kvantitatív output esetén klasszifikációt hajtunk végre. Ugyanakkor bizonyos esetekben a klasszifikációra is tekinthetünk regresszióként, ha osztály valószínűségeket prediktálunk konkrét osztály helyett.

## 2.4. Klasszifikáció

Klasszifikáció során tehát kvalitatív válaszokat prediktálunk, amelyek egy véges halmazból vesznek fel értékeket. Ilyen módszer például a logisztikus regresszió, a lineáris diszkriminancia-analízis, a KNN, a döntési fa, a random forest, a boosting és a support vector machine. Felmerülhet a kérdés, hogy miért nem alakítjuk át a válaszokat numerikus értékké és alkalmazunk rajtuk regressziós módszereket. Ez azért nem egy járható út, mivel azt feltételezi, hogy létezik egy bizonyos sorrend a kategóriák között. Például, ha az A, B, C kategóriákat 1, 2, 3 számoknak feleltetjük meg, akkor azt feltételezzük, hogy ugyanakkora a "távolság" A és B között, mint a B és C között. Ugyanakkor megválaszthatjuk az átalakítást másféleképpen is, pl. A=3, B=2 és C=1. Erre viszont különböző modellt kapnánk lineáris regresszióval. Ez a módszer csak akkor működik, ha a kategóriák sorba állíthatók (pl. kis, közepes, nagy) és a kategóriák közötti "különbség" azonos vagy ha csak két kategória van - bináris/kétszintű válaszok esetén a két lehetséges kódolás bármelyikére ugyanazt az eredményt kapjuk.

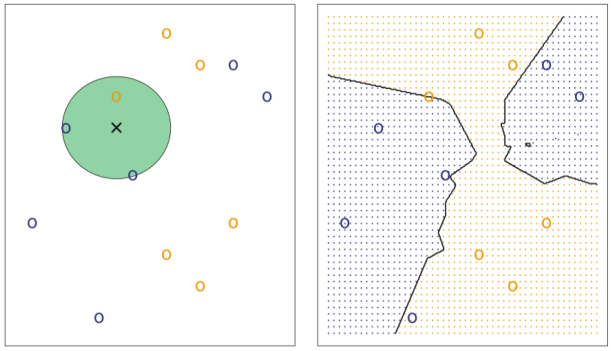
## 2.5. K-Nearest Neighbors

### KNN klasszifikáció

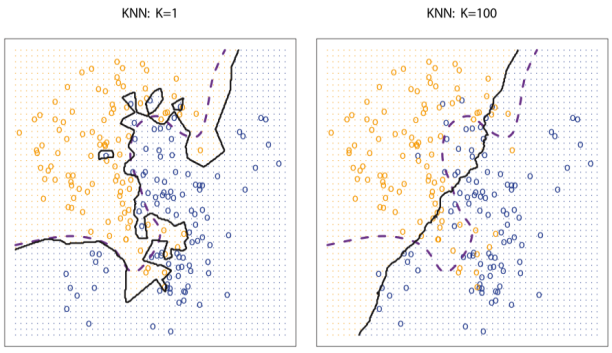
A K-Nearest Neighbors (KNN) klasszifikáció egy olyan módszer, amely a megfigyelések szomszédos pontjainak outputjából jósolja meg, hogy a megfigyelések melyik kategóriába tartoznak. A KNN Y X szerinti feltételes valószínűségét becsli és osztályozza a megfigyeléseket a legmagasabb becsült valószínűség szerint:

$$Pr(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in N_o} I(y_i = j),$$

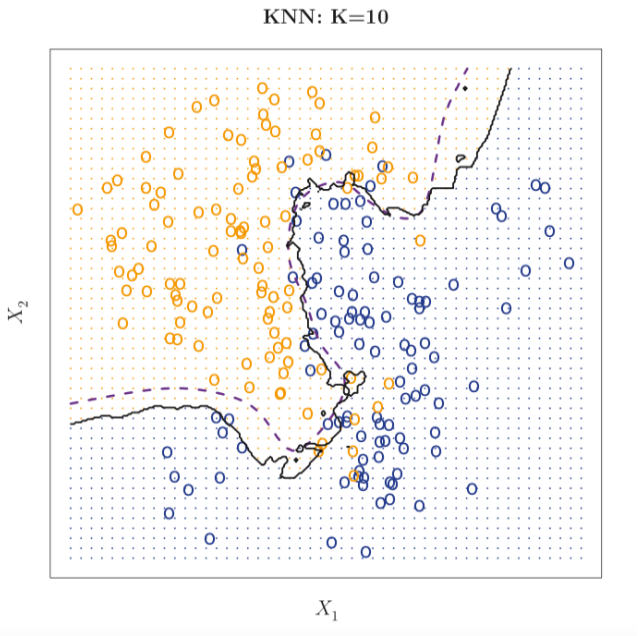
ahol $Pr(Y = j|X = x_0)$ jelöli annak a valószínűségét, hogy a megfigyelés a j kategóriába esik feltéve, ha a szóban forgó megfigyelés éppen $x_0$, K a számításba vett szomszédok száma (egész szám), $N_0$ $x_0$-hoz legközelebb levő K db szomszédos megfigyelés halmaza, I pedig egy indikátor változó, amelynek 1 az értéke, ha y a j kategóriába tartozik és 0 ellenkező esetben. Ezután a KNN abba a kategóriába rendeli a megfigyelést, amelynek legnagyobb a valószínűsége. A KNN működését a 1 ábra szemlélteti.
K megválasztása nagy hatással van az osztályozásra. K=1 esetén a döntési határ túl flexibilis és túlérzékeny a mintázatra. K növekedésével a módszer flexibilitása csökken és a lineárist egyre inkább megközelítő döntési határt produkál (2 és 3 ábra).



1. ábra. KNN megfigyelés K = 3 esetén. A próba megfigyelést, amelyre meg szeretnénk határozni, hogy melyik kategóriába tartozik, a fekete x jelöli (bal oldal). A tanítóhalmaz 6 db kék és 6 db narancssárga osztályhoz tartozó adatot tartalmaz. A kör a kijelölt ponthoz legközelebb eső 3 szomszédot veszi körül, két kék és egy narancssárga pontot. A leggyakrabban előforduló kategória a kijelölt pont K db szomszédja között tehát a kék, így a KNN azt prediktálja, hogy a kijelölt pont is a kék osztályhoz tartozik. A jobb oldali képen látható a döntési határ fekete vonallal, amely elhatárolja, hogy a teszt pontok mely kategóriába lesznek sorolva annak függvényében, hogy a narancssárga vagy a kék területen helyezkednek el.




2. ábra. Döntési határok összehasonlítása K=1 és K=100 esetén (fekete folytonos vonalak) az elméleti modellel (szaggatott vonal). A döntési határ az első esetben túl flexibilis, a második esetben pedig nem elég flexibilis.




3. ábra. K=10 esetén prediktált döntési határ (folytonos fekete vonal) összehasonlítása az elméleti határral (szaggatott vonal).

#### KNN regresszió

A KNN regresszió hasonlóan működik, mint a KNN klasszifikáció, viszont kategóriák helyett numerikus értékeket prediktál minden megfigyelés esetén:

$$\hat{f}(x_0) = \frac{1}{K} \sum_{i \in N_o} y_i.$$

A KNN módszer egy nem-paraméteres módszer, tehát semmilyen feltételezést nem teszünk a döntési határ alakjáról.

### 2.6. Random Forest

## Hivatkozások