Felügyelt tanulás

A felügyelt tanulás esetén a példákhoz meg vannak adva a helyes osztálycímkék is. Azaz a példáink a kövezkező alakban írhatók le: $(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)$, ahol x_i a minta és y_i a hozzá tartozó osztálycímke. A feladat a még nem ismert példákhoz a hozzájuk tartozó osztálycímke megmondása a minták alapján. A mintákat jellemzőkkel (feature) írhatjuk le. A különböző minták jellemzőinek és osztálycímkéinek korrelációjából következtethetünk egy még ismeretlen minta osztályára. A jellemzők lehetnek binárisak (igen/nem), véges diszkrét értékűek, valamint valós értékűek.

Feladat: A példákban jelen lávő rejtett minták alapján egy $X \rightarrow Y$ leképezés, amely várhatóan a tanítás során nem látott minta esetén is helyes eredményt ad.

Felügyelt módszerek fajtái:

Egyosztályos tanulás Egyosztályos tanulás esetén csak az adott osztályhoz tartozó minták adottak. A feladat eldönteni, hogy a még nem látott minta bele tartozik-e ebbe az osztályba.

Kétosztályos tanulás A kétosztályos (bináris) tanulás estetén az ismert minták a két osztály valamelyikébe taroznak. Az ismeretlen mintákról pedig el kell dönteni, hogy melyik osztályba tartoznak a kettő közül.

Többosztályos tanulás Többosztályos tanulás esetén $N > 2$ véges számú osztálycímkénk van. Minden N osztályos tanulás visszavezethető n db bináris osztályozási feladatra.

Regresszió Függvénytanulás (mind a jellemzők, mind az osztálycímkék valósak)

Azaz osztályozás esetén *diszkriminatív* megközelítésben keresnünk kell az n dimenziós térben (ahol n a jellemzők száma) egy olyan felületet, amely az egyes osztályokat szeparálja egymástól. *Generatív* megközelítésben pedig minden osztályhoz hozzárendelünk egy döntési függvényt, amely azt mondja meg, hogy az adott minta milyen mértékben tartozik az adot osztályhoz.

Minták közötti távolságmértékek: Ha a minták n jellemzővel vannak reprezentálva.

- Euklideszi távolság: $d(i,j) = \sqrt{\sum_k (x_i[k] - x_j[k])^2} \quad , k = 1, \ldots, n$
- Manhattan távolság: $d(i,j) = \sum_k |x_i[k] - x_j[k]| \quad , k = 1, \ldots, n$
- Cosinus távolság: $d(i,j) = \frac{x_i \times x_j}{||x_i|| \cdot ||x_j||}$

Osztályozók kiértékelésének formái

- Tévesztési (confusion) mátrix

		Aktuális feltétel	
		A feltétel teljesül	A feltétel nem teljesül
Teszt eredmény	Pozitív	A feltétel teljesül+pozitív teszt = TP (True Positives)	A feltétel nem teljesül+pozitív teszt = FP (False Positives)
	Negatív	A feltétel teljesül+negatív teszt = FN (False Negatives)	A feltétel nem teljesül+negatív teszt = TN (True Negatives)

- TPR (True Positive Rate): TP/(összes pozitívok száma)
- FPR (False Positive Rate): FP/(összes negatívok száma)
- FNR (False Negative Rate): FN/(összes pozitívok száma)
- TNR (True Negative Rate): TN/(összes negatívok száma)
- Sensitivity=TP/(TP+FN) :a pozitívak helyesen felismert aránya
- Specificity=TN/(FP+TN) :a negatívak helyesen felismert aránya
- Accuracy=(TP+TN)/(TP+TN+FP+FN): a helyesen felismert adatok aránya
- Recall=Sensitivity
- Precision= TP/(TP+FP) :a pozitívként felismertek hanyad része volt valóban pozitív
- F-measure: 2*Precision*Recall/(Precision+Recall) :harmonikus közepe a precison és recall értékeknek

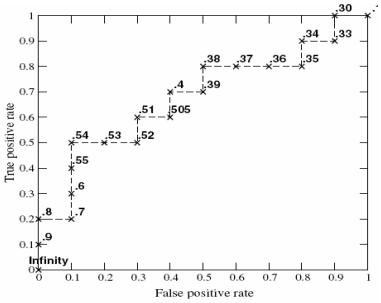
A Recall, Precision és F-measure mértéket egyszerre csak 1 osztályra, vagy osztályok egy részére érdemes alkalmazni, ugyanis, ha az összes osztályra alkalmazzuk egyszerre, akkor az Accuracy értéket kapjuk meg.

Ha adunk a mintákhoz egy rendezést, rangsort a pozitív osztályhoz való tartozás alapján, és definiálunk egy operációs küszöböt, ami egy érték, amely feletti pontérékkel rendelkező adat az 1-es (pozitív) osztályhoz tartozik, alatta pedig a negatívba. Minden egyes küszöbhöz kiszámíthatjuk a TPR,FPR értéket. A TPR FPR teret ROC (Receiver Operating Characteristics) térnek nevezzük. A ROC görbe előnye, hogy nem érzékeny az osztályok kiegyensúlyozatlanságára, az osztályok közötti adatszám arányára.

Az AUC (Area Under an ROC Curve) a ROC görbe alatti terület. Az AUC azt fejezi ki, hogy mennyire soroljuk a pozitív mintákat a negtívak elé a tesztelés folyamán.

A ROC görbe rajzolásánál szokás még a Specificity Sensitivity teret használni, valamit ha jelentősen nagyobb a negatív példáink száma, mint a pozitívaké és számunkra nem lényeges a negatív osztályozásá pontossága , akkor a Recall Precision ROC görbére szokás AUC-ot számolni.

Inst#	Class	Score	Inst#	Class	Score
1	p	.9	11	p	.4
2	p	.8	12	n	.39
3	n	.7	13	p	.38
4	p	.6	14	n	.37
5	p	.55	15	n	.36
6	p	.54	16	n	.35
7	n	.53	17	p	.34
8	n	.52	18	n	.33
9	p	.51	19	p	.30
10	n	.505	20	n	.1



Egy másik lehetséges módszer az osztályozó teljesítményének mérésére a hiba mértékének mérése (a következőkben az osztályozó által javasolt címkét az i-edik mintához $p(x_i)$, a helyes címkét pedig y_i fogja jelölni, n pedig a minták számát):

- Négyzetes hiba:
$$\sum_i (p(x_i) - y_i)^2 \quad i = 1, \ldots, n$$
- Átlagos négyzetes hiba (MSE):
$$\sum_i \frac{1}{n}(p(x_i) - y_i)^2 \quad i = 1, \ldots, n$$
- Középhiba (RMSE):
$$\sqrt{\sum_i \frac{1}{n}(p(x_i) - y_i)^2} \quad i = 1, \ldots, n$$

Példahalmaz felosztása a validáció szempontjából

- Tanuló és teszthalmaz (pl. 2/3 és 1/3 arányban). Problémát okozhat a túlillesztés és a túlzott általánosítás.
- Tanuló, ellenőrző és teszthalmaz (pl. 70% / 20% / 10%) arányban (ellenőrző (validation) halmaz: egy tanulómodell megfelelő paramétereinek hangolására, vagy több konkurens modell közül a legjobb kiválasztására)
- A teljes példahalmaz 2 v. 3 részre osztását többször végezzük el
- A felosztott (train, test, valid.) halmazok diszjunktak legyenek
- A felosztás véletlen mintavételezéssel (ha van olyan osztály, ami nagyon rosszul reprezentált, akkor rétegzett mintavételezést használunk (minden osztályból legyen tanító és tesztadat is))
- Bootstrap módszer: Véletlen mintavételezés visszatevéses módszerrel. Tanítóhalmaz: az N elemű teljes adathalmazból válasszunk ki visszatevéssel N elemet. Teszthalmaz: a maradék. Annak a valószínűsége, hogy egy elemet nem választunk be a tanítóhalmazba: $(1 - \frac{1}{N})^N = 0.368$, tehát, a teszthalmaz várhatóan az elemek 36.8%-át tartalmazza. A tanítóhalmazban egy-egy elem többször is megjelenhet

- Cross-Validation (kereszt validáció): A tanítóhalmazt N (legtöbbször 10) részre osztjuk fel, ebből 1 rész a teszthalmaz, a többi a tanító. Így N db tanítást és tesztelést kell elvégezni, ezek átlagos eredményével jellemezzük a módszert.
- Leave-one-out módszer: a cross-validation speciális esete, a teszthalmaz egy elemű. Előnye: a lehető legbővebb a tanítóhalmaz, és determinisztikus a módszer. Hátránya: a kiértékelések száma (ami a teljes mintahalmaz méret) nagy. Ezen kívül, a mintavétel nem rétegzett.

A $k-NN$ osztályozó

A $k-NN(k\ Nearest\ Neighbor)$ osztályozó esetén adottak a minták és a hozzájuk tartozó címkék. Ha egy még ismeretlen mintát szeretenénk a $k-NN$-el osztályozni, meg kell néznünk az ismeretlen mintához tartozó k db legközelebbi ismert minta címkéjét, majd azok alapján dönteni az ismeretlen címkéjéről. Ez történhet többségi szavazás alapon is, vagy lehet a szavazatokat a távolságok reciprokaival súlyozni. A $k-NN$ osztályozó egyik speciális formája az $1-NN$, amely esetben a legközelebbi szomszéd címkéjét kapja meg az ismeretlen minta.