Класификација ансамблом метода

Ненад Митић

Математички факултет
nenad@matf.bg.ac.rs

Идеја ансамбла метода

Основна идеја

- "Нема бесплатног ручка" - не постоји алгоритам који се најбоље понаша у свим могућим ситуацијама
- Комбиновање скупа модела који решавају (исти) оригинални проблем
- Циљ - добијање бољег глобалног модела
- Већа прецизност и поузданост процене у односу на сваки појединачни модел

Ансамбл метода

- "Ансамбл" метода - скуп метода које заједно "наступају" да би се добио бољи резултат
- *Condorcet* теорема жирија (пороте) (Marie Jean Antoine Nicolas de Caritat, маркиз de Condorcet (1743–1794))
- Нека група људи независно један од другог бира између две могућности од којих је само једна исправна, и нека је p вероватноћа да су изабрали исправну могућност. њихови гласови се комбинују по правилу већине, и нека M означава вероватноћу да је већина направила коректан избор. Ако је $p > 0.5$ тада $M \longrightarrow 1$ ако број гласања тежи ка бесконачности

Ансамбл метода

- Последица *Condorcet* теореме: гомила је паметнија од појединца под релативно слабим условима
- Сваки појединац мора да исправно суди са вероватноћом $p > 0.5$ (нешто мало боље од случајног нагађања)
- Сваки појединац одлуку доноси независно од осталих
- Недостаци: бинарна класификација + независност

Јаки и слаби класификатори

- Јаки класификатор: грешка класификације може да буде произвољно мала
- Слаби класификатор: класификатор који је нешто бољи од обичног случајног нагађања
- класификација ансамблом метода: уместо коришћења једног јаког класификатора формира се велики скуп слабих класификатора чији се излази комбинују у једно (финално) решење
- Према *Condorcet* теореми, ако се обезбеде одговарајући услови добиће се модел чија је грешка класификације произвољно близу нуле
- Додатни разлог за коришћење: лакше је направити више слабијих класификатора него један јак

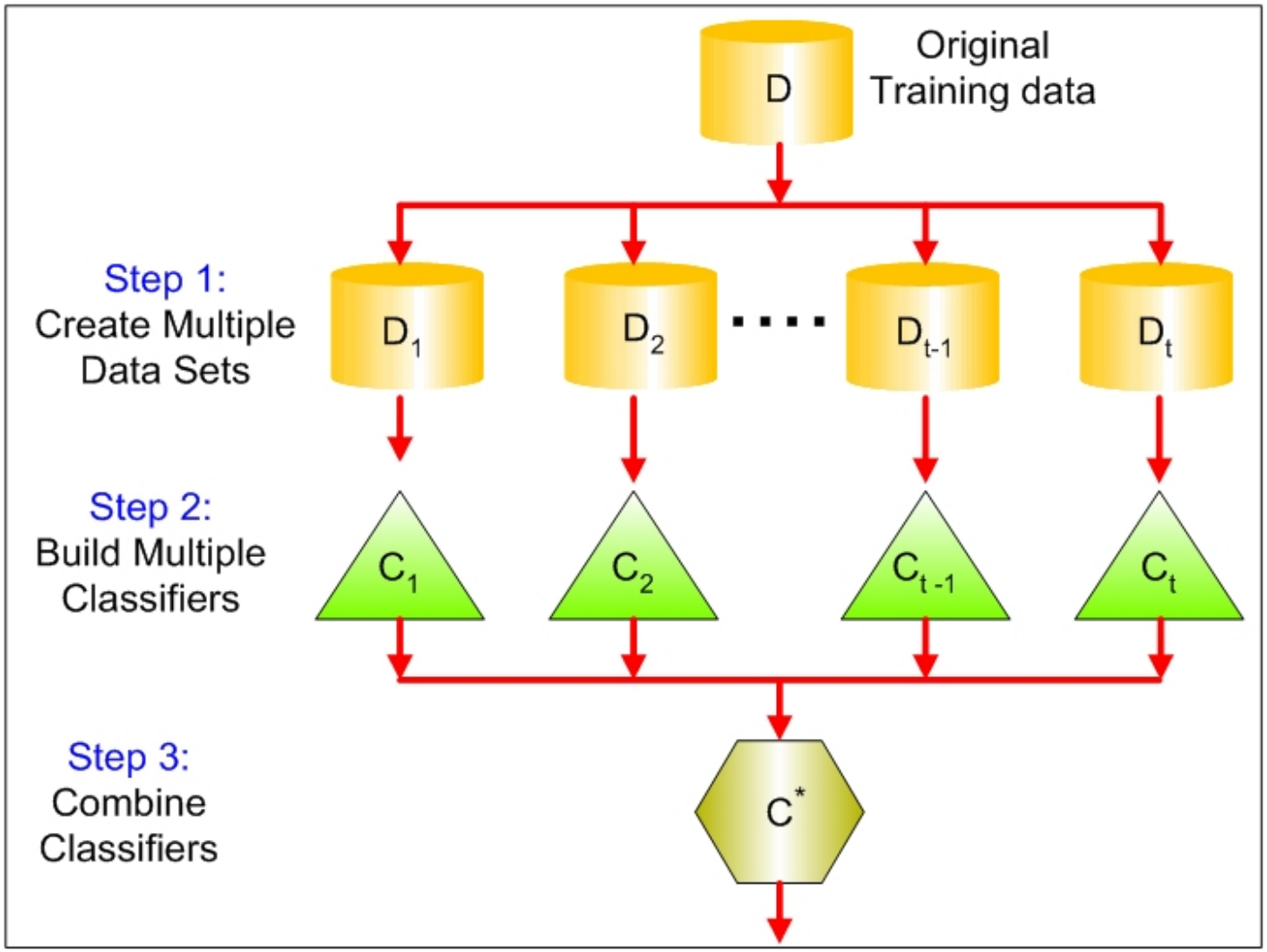
Грешка код ансамбла метода

1. Нека ансамбл чине 15 метода које имају грешку класификације од $\varepsilon = 0.3$
2. Ансамбл ће направити погрешно предвиђање ако више од половине елемената ансамбла има погрешно предвиђање
3. Због тога, грешка предвиђања ансамбла биће

$$\varepsilon_{ansambl} = \sum_{i=8}^{15} \binom{15}{i} \varepsilon^i (1-\varepsilon)^{15-i} = 0.05$$

што је знатно мање од грешке сваког појединачног класификатора

Логичка структура ансамбла метода



Методе за конструкцију ансамбла класификатора

- Променом скупа за тренинг
- Променом скупа улазних атрибута
- Променом скупа ознака класа
- Мењењем алгоритма за класификацију

Ансамбл метода боље ради са *нестабилним* класификаторима (дрвета одлучивања, неуронске мреже, класификатори засновани на правилима), тј. класификаторима који су осетљиви на незнатне промене у скупу за тренинг.

Промена скупа за тренинг

- Формира се више скупова за тренинг избором и почетног скупа податка на основу неког критеријума.
- Дистрибуција и избор елемената може да се мења при сваком избору
- Класификатор се формира применом (истог) алгоритма класификације на сваки од скупова за тренинг.
- Представници: алгоритми са додатним појачавањем (енг. *boosting*) и паковањем (енг. *bagging*)

Промена скупа улазних атрибута

- За сваки скуп података за тренинг бира се подскуп улазног скупа атрибута.
- Избор може бити случајан, али на основу датих директива
- Показано је да приступ ради јако добро у случају да улазни скуп садржи редундантне податке.
- Представници: насумична шума (енг. *Random forest*)

Промена скупа ознака класа

- Користи се када је скуп класа довољно велики
- Тренинг скуп се трансформише у бинарни проблем (0/1) случајним груписањем класа у два дисјунктна скупа
- Узастопним груписањима постиже се ефекат ансамбла; при тестирању ако класификатор предвиди класу , тада сви класификатори у групи којој приапада добијају један глас и обратно.
- Класа која добије највише гласова се додељује тест примеру
- Пример: кодирање излаза са отклањањем грешака

Промена алгоритма за класификацију

- Неки класификациони алгоритми дају различите моделе у примени на исте податке (нпр. неуронске мреже у случају промене топологије или почетних тежина за везе између неурона
- На пример, ансамбл метода са дрветима одлучивања може да се конструише тако што се укључи случајност у процедуру раста дрвета (уместо бирања најбољег атрибута за поделу у сваком чвору, случајно може да се бира један од најбољих n атрибута за поделу.

Паковање

Паковање (*Bagging, Bootstrap AGGregatING*) је техника која формира податке за тест узастопним узорковањем (са понављањем) података из почетног скупа у складу са унформном дистрибуцијом вероватноћа

- Сваки од тако формираних иницијалних (енг.*bootstrap*) скупова има исту кардиналност као и оригинални скуп
- Због избора са понављањем неки слогови могу да се јаве више пута, док неки могу и да се не појављују
- У просеку узоркованих скупова садржи 63% почетног скупа података јер је сваки узорак биран са вероватноћом $1-(1-1/N)^N$. Ако је N довољно велико вероватноћа конвергира ка $1-1/e \approx 0.632$

Паковање (алгоритам)

```
Neka je D skup ulaznih podataka i
k broj inicijalnih skupova

for i=1 to k do
    formiraj inicijalni uzorak Di velicine N
    Trenirati osnovni klasifikator Ci na skupu Di
end for
C*(x)=klasa koja je dobila najveci broj glasova
```

Паковање - пример

Нека су дати подаци за тренинг над којима је добијен следећи резултат класификације

x	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
y	1	1	1	-1	-1	-1	-1	1	1	1

На основу ових резултата подела слога може бити на $x \leq 0.35$ или на $x \leq 0.75$ што даје прецизност класификације 70%.

Применом методе паковања циљ је наћи скуп од 10 једноставних (слабих) класификатора који у ансамблу коректно класификују овај скуп. Сваки слаби класификатор за $x \leq K$ одређује класу $+1$ или -1 у зависности од тога која вредност даје најмању грешку, где је K одређено минимизацијом ентропије

Паковање - пример

x	0.1	0.2	0.2	0.3	0.4	0.4	0.5	0.6	0.9	1.0	x<=0.35==>y=1
y	1	1	1	1	-1	-1	-1	-1	1	1	x>0.35==>y=-1
x	0.1	0.2	0.3	0.4	0.5	0.8	0.9	1	1	1	x<=0.65==>y=1
y	1	1	1	-1	-1	1	1	1	1	1	x>0.65==>y=1
x	0.1	0.2	0.3	0.4	0.4	0.5	0.7	0.7	0.8	0.9	x<=0.35==>y=1
y	1	1	1	-1	-1	-1	-1	-1	1	1	x>0.35==>y=-1
x	0.1	0.1	0.2	0.4	0.4	0.5	0.5	0.7	0.8	0.9	x<=0.3==>y=1
y	1	1	1	-1	-1	-1	-1	-1	1	1	x>0.3==>y=-1
x	0.1	0.1	0.2	0.5	0.6	0.6	0.6	1	1	1	x<=0.35==>y=1
y	1	1	1	-1	-1	-1	-1	1	1	1	x>0.35==>y=1
x	0.2	0.4	0.5	0.6	0.7	0.7	0.7	0.8	0.9	1	x<=0.75==>y=-1
y	1	-1	-1	-1	-1	-1	-1	1	1	1	x>0.75==>y=1
x	0.1	0.4	0.4	0.6	0.7	0.8	0.9	0.9	0.9	1	x<=0.75==>y=-1
y	1	-1	-1	-1	-1	1	1	1	1	1	x>0.75==>y=1
x	0.1	0.2	0.5	0.5	0.5	0.7	0.7	0.8	0.9	1	x<=0.75==>y=-1
y	1	1	-1	-1	-1	-1	-1	1	1	1	x>0.75==>y=1
x	0.1	0.3	0.4	0.4	0.6	0.7	0.7	0.8	1	1	x<=0.75==>y=-1
y	1	1	-1	-1	-1	-1	-1	1	1	1	x>0.75==>y=1
x	0.1	0.1	0.1	0.1	0.3	0.3	0.8	0.8	0.9	0.9	x<=0.05==>y=-1
y	1	1	1	1	1	1	1	1	1	1	x>0.05==>y=1

Паковање - пример

Korak x=	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
1	1	1	1	-1	-1	-1	-1	-1	-1	-1
2	1	1	1	1	1	1	1	1	1	1
3	1	1	1	-1	-1	-1	-1	-1	-1	-1
4	1	1	1	-1	-1	-1	-1	-1	-1	-1
5	1	1	1	-1	-1	-1	-1	-1	-1	-1
6	-1	-1	-1	-1	-1	-1	-1	1	1	1
7	-1	-1	-1	-1	-1	-1	-1	1	1	1
8	-1	-1	-1	-1	-1	-1	-1	1	1	1
9	-1	-1	-1	-1	-1	-1	-1	1	1	1
10	1	1	1	1	1	1	1	1	1	1
Zbir	2	2	2	-6	-6	-6	-6	2	2	2
Znak	1	1	1	-1	-1	-1	-1	1	1	1
Koretna klasa	1	1	1	-1	-1	-1	-1	1	1	1

Појачавање

Појачавање (*Boosting*) је техника адаптивне промене дистрибуције тренинг података у зависности од претходних грешака класификације

- Иницијално, сваком од N слогова се додели једнака тежина
- Тежина се мења на крају сваког циклуса - тежина слогова који су погрешно класификовани се повећава, а тачних смањује
- Финални класификатор комбинује гласове свих класификатора у циклусу

Један од најпопуларнијих алгоритама је *AdaBoost* (*Adaptive Boost*). Видети SPSS modeler Algorithm Guide

Насумична шума

Посебно конструисана метода за ансамбл дрвета одлучивања

- Конструише се више дрвета
- Ансамбл непоткресаних дрвета одлучивања
- Сваки основни класификатор конструише нови вектор атрибута из оригиналних података
- Свако дрво користи случајни вектор генерисан са фиксном дистрибуцијом расподеле
- Коначан резултат се добија гласањем (преко свих дрвета у шуми)

Насумична шума

Случајни вектор може да се формира на више начина

- Случајно се бира F улазних карактеристика за поделу у сваком чвору дрвета (*ForestRI - Random Input selection*)
- У сваком чвору формира се нови атрибут на основу случајно изабраних Л атрибута. Нови атрибут је линеарна комбинација изабраних атрибута са коефицијентима генерисаним униформном дистрибуцијом у интевалу $[-1, 1]$. Затим се у сваком чвору генерише F таквих случајно комбинованих атрибута, од којих се најбољи бира за поделу у чвору (*ForestRC*)
- У сваком чвору најбоља подела се добија случајним избором између F најбољих атрибута уместо између свих атрибута