ÚLOHA 7
Naivní bayesovský klasifikátor

Zadáno na cvičení: 8
Mezní termín: 6.12. 2017
Maximální počet bodů: 15
Nepovinná úloha

Zadání

Stáhněte si archiv *bayes.zip* ze stránky *Bayes*. Archiv obsahuje tyto soubory:

- *bayesClassifyDocuments.m* – hlavní skript pro úlohu klasifikace dokumentů
- *bayesExample.m* – klasifikace studentů z úlohy na logistickou regresi
- *bayesGetProbs.m*[1] – odhad pravděpodobností tříd Bayesovského klasifikátoru.
- *bayesMultiExample.m* – hlavní skript pro úlohu klasifikace číslic.
- *bayesPredict.m* – predikce pro NBK
- *checkProbabilities.m* – Kontrola, že pravděpodobnosti příznaků za podmínky klasifikace do jednotlivých tříd sčítají do jedné
- *crossValidation.m* – křížová validace pro využití stejných dat pro trénování i testování
- *data/data1.txt* – vstupní data pro klasifikaci studentů
- *data/data2.mat* – vstupní data pro klasifikaci číslic v binárním formátu MATLABu.
- *displayData.m* – vizualizace dat
- *equidistantFeatureTransform.m*[2] – transformace příznaků na podintervali tak, aby všechny intervaly měly stejný rozsah
- *equisizedFeatureTransform.m* – transformace příznaků na podintervaly tak, aby všechny intervaly obsahovaly stejný počet prvků.
- *getFeatureClassProb.m*[1] – výpočet pravděpodobnosti příznaku za podmínky dané třídy
- *loadDocuments.m* – načtení klasifikovaných dokumentů do matice
- *plotData.m* – vizualizace dat
- *trainBayes.m*[1] – učící funkce NBK

Soubory označené [1] budete doplňovat v první části.
Soubory označené [2] budete doplňovat ve druhé části.

1 Naivní bayesovský klasifikátor

V této úloze budete programovat NBK a testovat jeho úspěšnost na předchozích úlohách.

Data

V první části budete klasifikovat text. Jedná se o internetové zprávy a jejich klasifikaci do dvaceti tříd. Data a informace o nich naleznete na `http://qwone.com/~jason/20Newsgroups/`

Data, která máte přiložená v archivu jsou už upravená do snadněji zpracovatelného formátu. Na každém řádku je trojice: index dokumentu index slova počet. Z tohoto formátu jsou funkcí *loadDocuments.m* načteny do matice, kde každý řádek o velikosti slovníku reprezentuje jeden dokument. Matice obsahuje četnosti jednotlivých slov v dokumentech.

Data jsou uložena ve složce *data*. Jsou rozdělena na trénovací a testovací část. každá část se skládá ze třech souborů:

1. .data – výše zmíněné trojice
2. .label – index třídy jednotlivých dokumentů
3. .map – mapování indexů tříd na názvy

Úkoly

1. **Naprogramujte trénování NBK** (v souboru *trainBayes.m*). Struktura natrénovaného modelu není pevně daná, ale v zásadě si do struktury model potřebujete uložit dvě datové struktury.
   (a) vektor pravděpodobností tříd (nebo četností dokumentů ve třídách)
   (b) Matici pravděpodobností příznaku za podmínky třídy (nebo četností příznaků ve třídě).
2. **Naprogramujte výpočet pravděpodobnosti slova za podmínky dané třídy** (funkce *getFeatureClassProb.m*).

   Po kroku 2 by měla kontrola pravděpodobností vypisovat všechny pravděpodobnosti 1. Funkce *checkProbs.m* posčítá podmíněné pravděpodobnosti přes všechna slova pro každou třídu (musí být 1). Tato funkce slouží pouze v ověření základních vlastností pravděpodobnosti.
3. **Naprogramujte MAP odhad pravděpodobností tříd** (funkce *bayesGetProbs.m*).

   Tip: Při výpočtu pravděpodobnosti příznaků za podmínky třídy celou rovnici zlogaritmujte a počítejte sumu logaritmů pravděpodobností místo součinu pravděpodobností. Pravděpodobnosti jsou totiž malá čísla a součin brzy podteče přesnost čísla v pohyblivé řádové čárce. Výsledná čísla už tedy nemusí být (a ani by neměli být) pravděpodobnostmi v pravém slova smyslu, ale měla by těm skutečným pravděpodobnostem být přímo úměrná.

   $$\arg\max_c logP(c_i) + \sum_j^a logP(a_j|c_i)$$
4. **Přidejte vyhlazování přičtením jedničky.**

   Ke každé četnosti přičtěte 1. Tak se zbavíte nulových pravděpodobností, čili i nikdy neviděné slovo má nenulovou pravděpodobnost.

Accuracy na klasifikaci dokumentů by měla být kolem 78% na testovacích datech.

2 Klasifikace s reálnými příznaky

V této části budete pomocí bayesovského klasifikátoru řešit předchozí klasifikační úlohy.

Úkoly

1. Implementujte rozdělení příznaků na stejně velké podintervaly (funkce *equidistantFeature-Transform.m*). vstupem je pole nebo matice příznaků a funkce musí kačdou hodnotu převést na one-hot vektor, odpovídající podintervalu, do kterého příznak spadá. Pokud tedy vstupem bude vektor:

$$\begin{pmatrix} 2 \\ 4 \\ 6 \\ 8 \\ 10 \end{pmatrix}$$

   a budeme chtít příznak rozdělit na dva podintervaly, výstupem bude:

$$\begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$$

   V aplikaci už je implementovaná podobná funkce, která rozdělí příznak na intervaly o stejném počtu prvků (*equisizedFeatureTransform.m*).

Po implementaci transformace můžete pustit skripty *bayesExample.m* a *bayesMultiExample.m*. Accuracy na první úloze by měla být zhruba 90%, na druhé úloze zhruba 84%. Můžete zkusit vyladit počet intervalů, abyste dosáhli lepších výsledků.