Klasifikace, Gaussian Mixture Models GMM

Jan Černocký, Mirko Hannemann, Karel Veselý FIT VUT Brno

V tomto cvičení si ukážeme prinicip statistických klasifikátorů založených na modelování dat pomocí gaussovského rozložení.

1 Klasifikace dvou tříd v 1-D

V prvním příkladu se budeme zabývat zjednodušeným problémem, kdy jednotlivé vstupy přiřazujeme do dvou tříd na základě jednorozměrných příznaků (výška, krevní tlak).

1.1 Klasifikace lidí na velké a malé

Vytvoříme sadu 100 trénovacích vzorků pro každou třídu. Vzorky získáváme z generátoru náhodných čísel s rovnoměrným rozložením omezeném na intervaly:

- velcí lidé: výška 170-210 cm
- malí lidé: výška 140-180 cm

```
%generate 2*100 training examples
data1 = rand(1,100) * (210-170) + 170; %big people
data2 = rand(1,100) * (180-140) + 140; %small people
figure; subplot 211; plot(data1,zeros(size(data1)),'or',data2,ones(size(data2)),'ob');
axis([130 220 -1 2]); subplot 212; hist([data1' data2'],20); axis([130 220 -1 20]);

%write data to file: first column height, second column class label '1'/'2'
ff = fopen('aux.txt','w');
fprintf (ff,'%f %d\n',[data1; ones(size(data1))]);
fprintf (ff,'%f %d\n',[data2; 2*ones(size(data1))]);
fclose (ff);
```

Vygenerovaná data zapíšeme do souboru. Dále si vytvoříme dvě sady 50 testovacích vzorků, generujeme ze stejného rozložení:

```
% generate 2*50 test examples
data1 = rand(1,50) * (210-170) + 170; %big
data2 = rand(1,50) * (180-140) + 140; %small
figure; subplot 211; plot(data1,zeros(size(data1)),'or',data2,ones(size(data2)),'ob');
axis([130 220 -1 2]); subplot 212; hist([data1' data2'],20); axis([130 220 -1 20]);

ff = fopen('aux1.txt','w'); %write to file
fprintf (ff,'%f %d\n',[data1; ones(size(data1))]);
fprintf (ff,'%f %d\n',[data2; 2*ones(size(data1))]);
fclose (ff);
```

Takto získaná data náhodně 'zamícháme' pomocí perl skriptu `bordelify.pl`, který spustíme v shellu:

```
perl bordelify.pl aux.txt > velcimali_train.txt
perl bordelify.pl aux1.txt > velcimali_test.txt
```

Nyní znovu načteme trénovací data ze souboru `velcimali_train.txt` a pro každou třídu natrénujeme model skládající se z jedné gaussovky.

Rozložení hustoty pravděpodobnosti gaussovky je dané vzorcem:

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{1}$$

V našem jednorozměrném případě tedy budeme odhadovat parametry μ (střední hodnota, angl. mean) a σ^2 (rozptyl, angl. variance):

```
% read data
[data, class] = textread ('velcimali_train.txt','%f %d');

% training of 1 gaussian on each class
data1 = data(find(class == 1)); %select all examples with class label '1'
data2 = data(find(class == 2));
m1 = mean(data1); v1 = std(data1)^2; %compute mean and variance
m2 = mean(data2); v2 = std(data2)^2;

% visualization of data and Gaussians.
figure; plot (data1, zeros(size(data1)),'or'); hold on;
plot (data2, zeros(size(data1)),'ob');
x = (100:240)';
px1 = gaus(x,m1,v1); plot (x,px1,'r');
px2 = gaus(x,m2,v2); plot (x,px2,'b'); hold off;
```

Střední hodnotu μ nám vrátí funkce `mean`, rozptyl σ^2 určíme přes standardní odchylku pomocí funkce `std`. Takto natrénovaný model zobrazíme spolu s trénovacími daty, pro vizualiaci použijeme funkci `gaus(data,mean,variance)`. Funkce `gaus` nám pro zvolené body `data` vrací hodnoty hustoty pravděpodobnosti gaussovského rozložení $p(x|\mu,\sigma^2)$. Podívejte se do souboru `gaus.m`.

Teď is načteme testovací množinu dat a pro každy prvek vypočteme klasifikační skóre tak, že v daném bodě pro obě třídy vyhodnotíme funkci `gaus`. Získané pravděpodobnosti vzájemně odečteme:

```
% load test data
[data, class] = textread ('velcimali_test.txt','%f %d');

% compute classification scores
scores = zeros(size(data));
for ii=1:length(data)
  s1 = gaus(data(ii),m1,v1); %probability class 1
  s2 = gaus(data(ii),m2,v2); %probability class 2
  scores (ii) = s1 - s2;     %classifier score
  %visualization lines: comment out after few examples
  hold off;  px1 = gaus(x,m1,v1); plot (x,px1,'r');
  hold on; px2 = gaus(x,m2,v2); plot (x,px2,'b');
  stem (data(ii),s1,'or','Markersize',12); stem (data(ii),s2,'ob','Markersize',12);
  hold off;  [data(ii) s1 s2 s1-s2]
  pause
end
```

Pro každý prvek jsme dostali skóre $s \in [-1, 1]$, a tak můžeme provést klasifikaci, a vyhodnotit celkovou úspěšnost pomocí funkce `eval_2_class`. Podívejte se do souboru `eval_2_class.m`.

```
% evaluation
eval_2_class (scores, class)
```

Úkoly

1. Je klasifikace vždy úspěšná? Jaká je příčina chyb?
2. Ve kterém bodě klasifikátor mění rozhodnutí z 'malých' na 'velké'? (angl. decision boundary)

3. Jakým způsobem klasifikátor rozhoduje o přiřazení prvků do tříd? Jsou rozhodnutí optimální?

4. Objevují se dva druhu chyb: `error12` and `error21`. Co znamenají?

5. Implementujte náhodné 'míchání' pomocí matlablovské funkce: `randperm`
   (řešení: `data_rand=data(randperm(size(data,2)));`)

6. Jaký je význam parametru σ^2?

7. Proč modelujeme třídy gaussovským rozložením, když víme, že data byla generovaná z rovnoměrného rozložení? Byla tato volba správná?

1.2 Klasifikace zdravých a nemocných lidí

Teď se podíváme na složitější příklad: Budeme chtít rozhodonout, jestli má osoba navštívit lékaře v závislosti na krevním tlaku. Podle krevního tlaku si osoby rozdělíme do tří skupin:

- 40-70: nízký krevní tlak
- 60-90: normální krevní tlak
- 80-170: vysoký krevní tlak

Problém je v tom, že k doktorovi by měli jít lidé s příliš nízkým i příliš vysokým tlakem: hustota rozložení nemocných lidí je komplexnější, bude se skládat ze dvou oddělených intervalů se dvěma vrcholy (modusy: odtud multi-modální rozložení). Vygenerujeme trénovací data (500) a testovací data (500) stejným způsobem jako v předchozím příkladu.

```
%generate training data
data1 = rand(1,300) * (90-60) + 60; %normal blood pressure 60-90
data2 = rand(1,100) * (70-40) + 40; %low blood pressure 40-70
data2 = [data2 rand(1,100) * (170-80) + 80;] %plus high blood pressure 80-170
figure; subplot(211); hold on;
plot (data1, ones(size(data1)),'bo');
plot (data2, zeros(size(data2)),'ro');
hold off; title('Train set');
%write data with class labels to file
ff = fopen('aux2.txt','w');
fprintf (ff,'%f %d\n',[data1; ones(size(data1))]);
fprintf (ff,'%f %d\n',[data2; 2*ones(size(data2))]);
fclose (ff);

% generate test data ...
data1 = rand(1,300) * (90-60) + 60;
data2 = rand(1,100) * (70-40) + 40;
data2 = [data2 rand(1,100) * (170-80) + 80;]
subplot(212); hold on;
plot (data1, ones(size(data1)),'bo');
plot (data2, zeros(size(data2)),'ro');
hold off; title('Test set');
ff = fopen('aux3.txt','w');
fprintf (ff,'%f %d\n',[data1; ones(size(data1))]);
fprintf (ff,'%f %d\n',[data2; 2*ones(size(data2))]);
fclose (ff);
```

Podívejte se na grafy rozdělení dat se dvěma modusy u třídy nemocných lidí. Trénovací data opět zamícháme:

```
perl bordelify.pl aux2.txt > zdravinemocni_train.txt
perl bordelify.pl aux3.txt > zdravinemocni_test.txt
```

Teď si natrénujeme klasifikátor se dvěma jedno-gaussovkovými modely tříd, to stejné jsme dělali v minulém příkladu:

```
% read data
[data, class] = textread ('zdravinemocni_train.txt','%f %d');
% training of 1 gaussian for each class
data1 = data(find(class == 1)); %select data points from class 1
data2 = data(find(class == 2));
m1 = mean(data1); v1 = std(data1)^2;
m2 = mean(data2); v2 = std(data2)^2;
% visualization of data and Gaussians.
figure; plot (data1, zeros(size(data1)),'or'); hold on;
plot (data2, zeros(size(data2)),'ob');
x = (0:200)';
px1 = gaus(x,m1,v1); plot (x,px1,'r');
px2 = gaus(x,m2,v2); plot (x,px2,'b'); hold off;
```

Podívejte se na odhadnutá rozdělení pro obě třídy, vypadají vpořádku? Pro každý prvek testovací množiny spočítáme skore a vyhodnotíme úspěšnost klasifikátoru:

```
% testing and evauation
[data, class] = textread ('zdravinemocni_test.txt','%f %d');
scores = zeros(size(data));
for ii=1:length(data)
  s1 = gaus(data(ii),m1,v1); %probability for class 1
  s2 = gaus(data(ii),m2,v2); %probability for class 2
   scores (ii) = s1 - s2;     %classification score
end
eval_2_class (scores, class)
```

Úkoly

1. Dostali jsme dobrý výsledek? Proč dostáváme víc `error12` než `error21`?
2. Kde je chyba? Co musíme udělat, aby se zvýšila úspěšnost klasifikace?

1.3 Multi-modální klasifikace zdravých a nemocných lidí

Abychom mohli natrénovat lepší model pro multi-modální data (třída 2 má dva modusy), použijeme pro každou třídu dvě gaussovky. Skupiny gaussovek budeme nazývat *směsi* (angl. mixture), jednotl. gaussovky budeme nazývat *komponenty* (angl. component). Přidáme tedy dvakrát po jedné komponentě a všem gaussovkám přidělíme váhu $w = 0.5$ (musí platit, že součet vah ve směsi je 1).

Nyní potřebujeme nějak rozhodnout, která data náleží ke které komponentě. Toto uděláme manuálně pomocí prahu uprostřed dat. Je-li hodnota větší než práh, přiřadíme data druhé komponentě, jinak přiřadíme první komponentě.

Celkové skóre směsi M získáme tak, že sečteme váhované skóre všech komponent:

$$p(x|M) = \sum_{c=1}^{C} w_c p(x|\mu_c, \sigma_c^2) \ ; \ \sum_{c=1}^{C} w_c = 1 \tag{2}$$

Načteme data a natrénujeme všechny 4 gaussovky:

```
% read data
[data, class] = textread ('zdravinemocni_train.txt','%f %d');
data1 = data(find(class == 1));
data2 = data(find(class == 2));

% training two gaussian components per class
% we manually divide the data, as we know 60-90 healthy, 40-70,80-170 ill
% estimate mean, variance for each component of each class
% hand setting of weights to 0.5
data = data1(find(data1 < 75)); m11 = mean(data); v11 = std(data)^2; w11=0.5;
data = data1(find(data1 >= 75)); m12 = mean(data); v12 = std(data)^2; w12=0.5;
data = data2(find(data2 < 75)); m21 = mean(data); v21 = std(data)^2; w21=0.5;
data = data2(find(data2 >= 75)); m22 = mean(data); v22 = std(data)^2; w22=0.5;

% visualization of data and Gaussians.
figure; plot (data1, zeros(size(data1)),'or'); hold on;
plot (data2, zeros(size(data2)),'ob');
x = (0:200)';
px1 = w11*gaus(x,m11,v11)+w12*gaus(x,m12,v12); plot (x,px1,'r');
px2 = w21*gaus(x,m21,v21)+w22*gaus(x,m22,v22); plot (x,px2,'b');
hold off;
```

Podívejte se na výsledná rozložení pravděpodobnosti, jsou bimodální. Popisuje tento model data lépe? Abychom zjistili jestli je nový model vhodnější pro klasifikaci, vyhodnotíme úspěšnost:

```
% load testing data
[data, class] = textread ('zdravinemocni_test.txt','%f %d');
% compute classification scores, evaluation of 1d classifier with 2 gaussians per class
scores = zeros(size(data));
for ii=1:length(data)
  s1 = w11*gaus(data(ii),m11,v11)+w12*gaus(data(ii),m12,v12); %probability of class 1
  s2 = w21*gaus(data(ii),m21,v21)+w22*gaus(data(ii),m22,v22); %probability of class 2
  scores (ii) = s1 - s2; %classification score
end
eval_2_class (scores, class)
```

Úkoly

1. Dosáhli jsme lepší úspěšnosti? Proč?
2. Kde je hlavní problém? Co můžeme zlepšit?
3. Je manuální rozdělení dat na dvě části vhodné pro obě třídy?
4. Zvolili jsme vhodné váhy komponent?

2 Klasifikace dvou tříd v n-D

2.1 Klasifikace pohlaví

Zkusíme použít stejný postup řešení na jeden složitější problém – problém rozpoznávání pohlaví `;-)`. V předchozích příkladech jsme klasifikovali uměle generovaná data, teď budeme klasifikovat 'opravdová' data. Na základě hlasu v nahrávce budeme klasifikovat na *muže* a *ženy*.

Máme přípravenou množinu audio souborů ve formátu `*.raw`, které po načtení převedeme na MFCC koeficienty (Mel Frequency Cepstral Coefficients). Pokud nevíte, jak se počítají, podívejte se na přednášku *"3: Předzpracování řeči, tvorba řeči, cepstrum"*, nebo se zeptejte někoho, kdo to ví...

Matlabovský skript `raw2mfcc.m` si nejprve 'zjistí' jména všech souborů `raw` ve složce, a pak pro každý soubor vypočítá MFCC koeficienty funkcí `mfcc.m`. Argumentem funkce `raw2mfcc` je jméno adresáře se soubory `*.raw`, výstupem funkce je cell array, kde každý cell obsahuje matici příznaků pro daný soubor:

```
%Read all the training and test data into cell-arrays
train_m = raw2mfcc('data/male/train');
train_f = raw2mfcc('data/female/train');
[test_m files_m] = raw2mfcc('data/male/test');
[test_f files_f] = raw2mfcc('data/female/test');

% For training, we do not need to know which frame come from which training segment.
% So, for each gender, concatenate all the training feature matrices into single matrix
train_m=cell2mat(train_m);
train_f=cell2mat(train_f);
```

Pomocí funkce `cell2mat` sloučíme matice trénovacích dat, podívejte se do nápovědy funkce `cell2mat`.
Pro každý řečový rámec jsme získali vektor 12 MFCC koeficientů, který odpovídá nějakému bodu v 12-dimenzionálním prostoru. Uvnitř tohoto prostoru budeme provádět klasifikaci.
Zobrazíme si 100 náhodně vybraných bodů ve 3D, použijeme dimenze 2,6 a 9. Jsou třídy zřetelně oddělené?

```
%show 100 randomly chosen points in 3D
ii = randperm(4000); ii = ii(1:100);
figure; plot3(train_m(ii,2),train_m(ii,6),train_m(ii,9),'ob'); grid; axis equal;
hold on; plot3(train_f(ii,2),train_f(ii,6),train_f(ii,9),'or'); hold off;
```

2.2 Klasifikace s použitím jedné gaussovky na třídu

Podobně jako v prvním příkladu natrénujeme dva modely s jednou gaussovkou.
Funkce hustoty pravděpodobnosti pro vícedimenzionální gaussovku (multivariate gaussian) je dána následující rovnící:

$$\mathcal{N}(\vec{x}|\vec{\mu},\Sigma) = \frac{1}{(2\pi)^{D/2}\cdot|\Sigma|^{1/2}}\cdot e^{\left\{-\frac{1}{2}(\vec{x}-\vec{\mu})^T\cdot\Sigma^{-1}\cdot(\vec{x}-\vec{\mu})\right\}} \tag{3}$$

kde D je dimenzionalita dat, $\vec{\mu}$ je vektor středních hodnot a Σ je kovarianční matice.
Obě gaussovky budou mít *diagonální* kovarianční maticí. Gaussovka s diagonální kovarianční maticí modeluje jen variabilitu, která je rovnoběžná s osami souřadnic. Neumožňuje tedy libovolné 'natočení' gaussovky. To nám nevadí, jsou-li dimenze dat vzájemně dekorelované.

Odhadneme parametry gaussovek a otestujeme klasifikátor na první mužské testovací promluvě `test_m{1}`. Pro každý rámec získáme skore (likelihood) obou modelů (gaussovek):

```
l_m = gaus(test_m{1}, mean(train_m), var(train_m, 1)); %likelihood of male model
l_f = gaus(test_m{1}, mean(train_f), var(train_f, 1)); %likelihood of female model
% Plot the frame-by-frame likelihoods obtained with the two models
figure; plot(l_f, 'r'); hold on; plot(l_m, 'b');
hold off; xlabel('frame'); ylabel('likelihood');
```

Moc toho nevidíme, zkusíme ještě zobrazit posteriorní pravděpodobnost a hodnoty log-likelihood:

```
% Plot frame-by-frame posteriors
figure('Name','diagonal covariance');
subplot 211; plot(l_m./(l_m+l_f), 'b'); hold on; plot(l_f./(l_m+l_f), 'r');
hold off; xlabel('frame'); ylabel('posterior');
% Plot frame-by-frame log-likelihoods
subplot 212; plot(log(l_m), 'b'); hold on; plot(log(l_f), 'r');
hold off; xlabel('frame'); ylabel('log-likelihood');
```

Naším úkolem ale není klasifikovat rámece, potřebujeme klasifikujeme celý řečový segment. S využitím předpokladu, že rámce byly genrovány nezávisle, můžeme vyjádřit sdruženou pravděpodobnost celé promluvy součinem $p(\vec{x}_1, \vec{x}_2, ..., \vec{x}_n) = p(\vec{x}_1) \cdot p(\vec{x}_2) \cdot ... \cdot p(\vec{x}_n)$. Násobení lze převést pomocí logaritmu na sčítání, takže klasifikační skóre promluvy můžeme získat takto:

```
sum(log(l_m))-sum(log(l_f))
```

Je-li skóre kladné, klasifikovali jsme muže, v opačném případě by se jednalo o ženský hlas.

Úkoly

1. Proč (log-) likelihood mužského a ženského modelu skoro stejně?
2. Které ze tří skóre vypadá nejlépe? (mužská promluva je modrá)
3. Co jsou to posteriory a jak se počítají? Které ze skóre je normalizované?
4. Co znamená "diagonální kovarianční matice"? Proč se používá?
5. Proč v `gaus.m` ověřujeme `if(size(COV) == size(MU))` ?

2.3 Gaussovský model s plnou kovarianční maticí

Zopakujeme celý postup s tím rozdílem, že model bude umožňovat i 'natočení' gaussovek díky plné kovarianční matici. Kovarianční matici odhadneme z dat pomocí matlabovské funkce `cov`:

```
%train models, eval. likelihoods
testdata=test_m{1};
l_m = gaus(testdata, mean(train_m), cov(train_m, 1));
l_f = gaus(testdata, mean(train_f), cov(train_f, 1));
%show graphs
figure('Name','full-covariance');
subplot 211; plot(l_f./(l_m+l_f), 'r'); hold on; plot(l_m./(l_m+l_f), 'b');
xlabel('frame'); ylabel('posterior');
subplot 212; plot(log(l_f), 'r'); hold on; plot(log(l_m), 'b');
xlabel('frame'); ylabel('log-likelihood');
%score
sum(log(l_m))-sum(log(l_f))
```

Úkoly

1. Zopakujte předchozí krok s promluvou od ženy: `testdata=test_f{1};`.
2. Jak teď zjistíme, zda se klasifikátor rozhodl pro muže nebo ženu?
3. Je lepší použít plnou nebo diagonální kovarianční matici?
4. Co znamená "plná kovarianční matice", jaké jsou její výhody a nevýhody?

2.4 Výpočet skóre pro všechny promluvy

Uložíme si střední hodnoty a kovarianční matice do proměnných, provedeme klasifikaci všech testovacích promluv:

```
mean_m = mean(train_m);
cov_m  = cov(train_m, 1);
mean_f = mean(train_f);
cov_f  = cov(train_f, 1);

test_set = test_m;
for ii=1:length(test_set)
  l_m = gaus(test_set{ii}, mean_m, cov_m);
  l_f = gaus(test_set{ii}, mean_f, cov_f);
  score(ii)=sum(log(l_m))-sum(log(l_f));
end
score
```

Úkoly

1. Zopakujte to samé pro množinu s ženami `test_set=test_f`.
2. Jaká je chybovost pro mužské a ženské mluvčí?

2.5 Klasifikace směsí gaussovek GMM

V posledním příkladu, podobně jako v případě s krevním tlakem, použijeme multi-modální model složený z více gaussovek. Směs gaussovek (Gaussian Mixture Model, GMM) vznikne sečtením několika váhovaných gaussovsek, přčemž součet vah musí být 1:

$$p(\vec{x}) = \sum_{c=1}^{C} w_c \cdot \mathcal{N}(\vec{x}|\vec{\mu_c}, \Sigma_c) \;\; ; \;\; \sum_{c=1}^{C} w_c = 1 \; , \tag{4}$$

kde C je počet komponent, w_c je váha komponeny, $\vec{\mu_c}$ je vektor středních hodnot a Σ_c je kovarianční matice. Více v přednášce *"9.Rozpoznávání HMM"*.

Na rozdíl od příkladu z krevním tlakem neznáme tvar rozložení dat – jsou ve 12 dimenzionálním prostoru a nevíme, kolik má rozložení vrcholů (modusů). Proto začneme se směsmi s jedinou kompomentou a postupně budeme počet zdvojnásobovat 'štípáním'. Trénování a 'štípání' budeme střídavě opakovat, dokud nedosáhneme požadovaný počet komponent.
Další problém je, že optimální parametry GMM modelu není možné najít v jediném kroku. Pro trénování GMM se používá iterativní algoritmus Expectation Maximization (EM).

Štípání komponent je implementované ve funkci `split_mix(weights, mus, sigmas)`, jednu iteraci trénování implementuje funkce `dgmixtrain(traindata, weights, mus, sigmas)`. Budeme trénovat GMM s diagonální kovarianční maticí, protože nemáme dostatek dat pro spolehlivý odhad plné kovarianční matice pro každou gaussovku.

Natrénujeme model "muž": nejprve jednu gaussovku, tu 'rozštípneme' na dvě, nakonec provedeme jednu iteraci EM algoritmu:

```
%train male model
splits=1; iters_per_split=1;
%start with single gaussian with diagonal covariance matrix
WW_m = [1]
MM_m = [mean(train_m)']
EE_m = [var(train_m, 1)']
% Function 'split_mix' doubles the number of gaussian components
% Function 'dgmixtrain' updates GMM parameters using single EM iteration
for ii=1:splits
  [WW_m, MM_m, EE_m] = split_mix(WW_m, MM_m, EE_m)
  for jj=1:iters_per_split
    [WW_m, MM_m, EE_m] = dgmixtrain(train_m', WW_m, MM_m, EE_m)
  end
end
```

Počet iterací mezi štípáním je možné nastavit napevno, rovněž celkový počet štípání nastavíme napevno. Tyto globální parametry nám budou ovlivňovat úspěšnost klasifikace.

Nyní stejným způsobem natrénujeme model ženského hlasu:

```
%train female model
WW_f = [1]
MM_f = [mean(train_f)']
EE_f = [var(train_f, 1)']
for ii=1:splits
  [WW_f, MM_f, EE_f] = split_mix(WW_f, MM_f, EE_f)
  for jj=1:iters_per_split
    [WW_f, MM_f, EE_f] = dgmixtrain(train_f', WW_f, MM_f, EE_f)
  end
end
```

Po natrénování modelů provedeme klasifikaci a vyhodnotíme úspěšnost:

```
test_set=test_m;
for ii=1:length(test_set)
  l_m = gmm_pdf(test_set{ii}', WW_m, MM_m, EE_m);
  l_f = gmm_pdf(test_set{ii}', WW_f, MM_f, EE_f);
  score(ii)=sum(log(l_m))-sum(log(l_f));
end
score
```

Úkoly

1. Zopakujte to samé se sadu ženských promluv `test_set=test_f`.
2. Jaká je úspěšnost klasifikace?
3. Který klasifikátor je lepší? GMM se dvěmi gaussovkami s diagonální kovarianční maticí, nebo GMM s jednou gaussovkou a plnou kovarianční maticí?
4. Zkuste zvyšovat počet komponent na 2-4-8-... a počet iterací EM algoritmu. Zvyšuje se úspěšnost klasifikace? Proč od určitého počtu gaussovek začne úspěšnost klasifikace klesat?
5. Podívejte se na 3-D vizualizaci GMM modelů.

2.6 Bonus: vizualizace GMM modelu

```
%select 1000 datapoints and show them
ii = randperm(4000); ii = ii(1:1000);
figure; axis equal; grid; hold on;
plot3(train_m(ii,2),train_m(ii,6),train_m(ii,9),'xb'); %train data male
plot3(train_f(ii,2),train_f(ii,6),train_f(ii,9),'xr'); %train data female
%show GMMs as ellipsoids
sc=0.5; %ellipse radius will be half of standard deviation
for ii=1:size(MM_m,2) %show male GMMgaussians (blue)
   [a b c] = ellipsoid(MM_m(2,ii),MM_m(6,ii),MM_m(9,ii), ...
                       sc*sqrt(EE_m(2,ii)), sc*sqrt(EE_m(6,ii)), sc*sqrt(EE_m(9,ii)));
   surf(a,b,c,ones(20));
end
for ii=1:size(MM_f,2) %show female gaussians (red)
   [a b c] = ellipsoid(MM_f(2,ii),MM_f(6,ii),MM_f(9,ii), ...
                       sc*sqrt(EE_f(2,ii)),sc*sqrt(EE_f(6,ii)),sc*sqrt(EE_f(9,ii)));
   surf(a,b,c,10*ones(20));
end
alpha 0.2; axis tight; hold off;
```