Počítačové zpracování řeči

Přednáška 5

DTW - pokračování

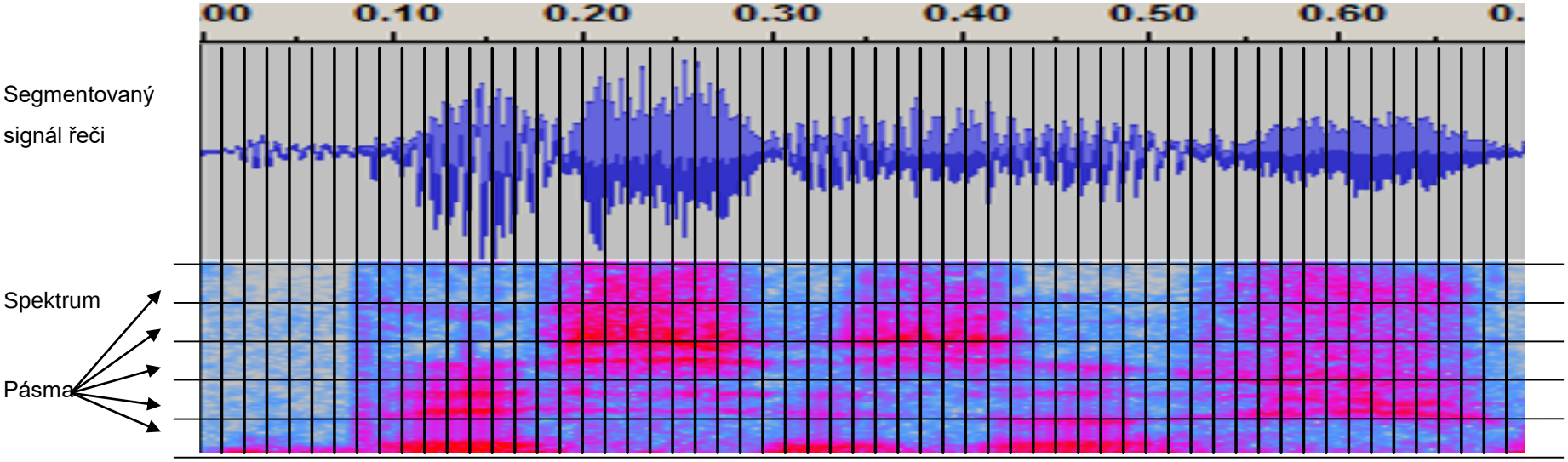
Lepší příznaky, efektivnější implementace

Zahrnutí spektrálních příznaků (1)

Idea: Různé hlásky mají různé spektrální charakteristiky.
Příznaky odvozené ze spektra tedy mohou zlepšit rozpoznávání.

Jednoduchý přístup zahrnutí spektrálních příznaků:

Pro každý frame spočítejme spektrum, rozdělme ho do několika pásem (též kanálů, angl. band nebo bin) a určeme energii v každém pásmu.

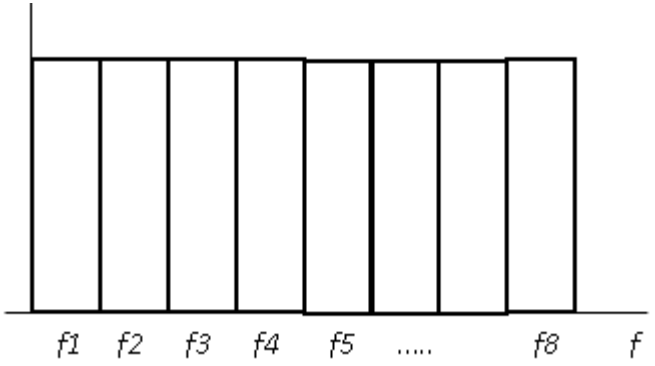


Spektrogram názorně ukazuje, jak se mění rozložení energie v čase a v pásmech.

Zahrnutí spektrálních příznaků (2)

Dosavadní zkušenost:

Vyzkoušeli jsme jednoduché rozdělení na 16 pásem a získání 16 příznaků – bez výraznějšího úspěchu (nepomohlo by ani více pásem)

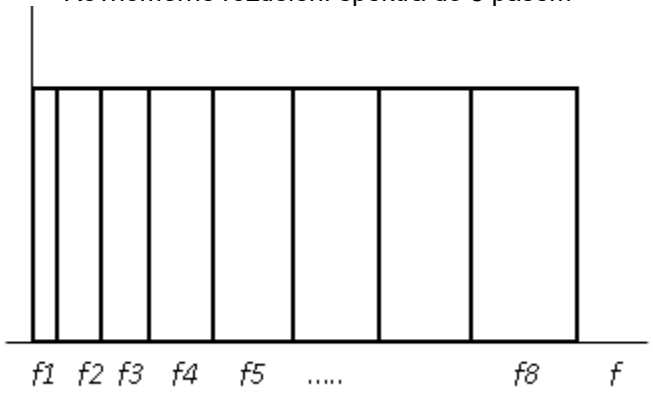


Rovnoměrné rozdělení spektra do 8 pásem

Nerovnoměrné rozdělení do pásem

Ukazuje se, že je lepší přizpůsobit pásma charakteristice lidského ucha, které mnohem citlivěji rozlišuje mezi nižšími než vyššími frekvencemi, tj. použijeme více užších pásem u nižších frekvencí, méně u vyšších.

Jak taková pásma vytvořit?

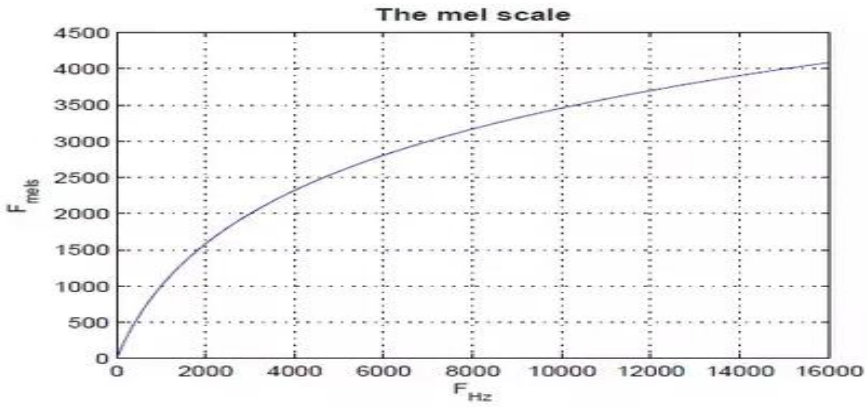


Nerovnoměrné rozdělení spektra do 8 pásem

Zahrnutí spektrálních příznaků (3)

Melová stupnice

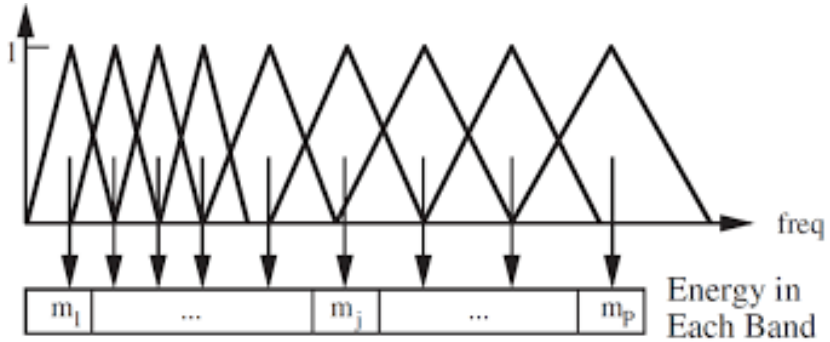
Empiricky odvozená křivka citlivosti rozlišení frekvencí v lidském uchu, převádí objektivní frekvenci v Hz na subjektivní stupnici v jednotkách zvaných Mel



Matematicky ji popisuje vztah:

$$\mathrm{Mel}(f) = 2595 \log\left(1 + \frac{f}{700}\right)$$

Podle ní vytvoříme pásma lépe odpovídající vlastnostem ucha. Pro počítačové zpracování navíc uspořádáme pásma tak, aby se částečně překrývala.



Obrázek znázorňuje 9 pásem, vytvořených podle mel stupnice. Pásma se částečně překrývají, tzn. každá frekvenční složka je započítána vždy do dvou sousedních pásem a to s váhovým koeficientem odpovídajícím trojúhelníkové funkci. Prakticky to znamená, že do jednoho pásma je složka je započítána s koeficientem q, do druhého s koeficientem 1-q.

Mel-spektrální příznaky (Fbank)

Mel-spektrální příznaky

se vypočítají podle postupu na předchozím slajdu:

1. FFT
2. Převod do mel-stupnice
3. Vytvoření K pásem a K trojúhelníkových filtrů
4. Energie signálu v pásmu = součet (energie jednotlivých frekvencí v pásmu * q)
   (q = váhový koeficient trojúhelníkové funkce)
5. Logaritmus energie

Tyto příznaky se často označují jako **Fbank** (banka trojúhelníkových frekv. pásem)

Kompletní program v Matlabu je uveden o 2 slajdy dále

Kepstrální příznaky (MFCC)

Kepstrální příznaky

Příznaky používané v profesionálních systémech rozpoznávání řeči

Vypočítají se pomocí inverzní Fourierovy transformace z Mel-spektrálních příznaků. (Protože hodnoty těchto koeficientů jsou reálné, IFFT se zjednoduší na tzv. diskrétní kosinovou transformaci DCT).

Označují se jako MFCC (Mel-Frequency Cepstral Coefficients)

Kompletní program v Matlabu je uveden na dalším slajdu.

Porovnání spektrálních, mel spektrálních a kepstrálních příznaků
na úloze z minulé přednášky (rozpoznávání metodou DTW, data z roku 2018):

Příznaky	**Úspěšnost [%]**
Spektrální příznaky (počet příznaků = 16)	79
Mel-spektrální příznaky (počet příznaků = 32)	82
Kepstrální příznaky (počet příznaků = 12)	89

Výpočet Fbank a MFCC příznaků

```
function [cep_coeff, mel_fbank] = computeFrameMFCC(frame, N, M, Fs)
    % vypočítá N příznaků typu Fbank a M příznaků typu MFCC pro jeden frame řeči
    % typické hodnoty N=26, M=12, Fs=16000, frame 25 ms = 400 vzorku
    % program volá 3 malé pomocné funkce: f2mel, mel2f, spread_mel
    m_low=0;                %spodní limit mel-stupnice
    m_top=f2mel(Fs/2);      %horní limit mel-stupnice
    mdiv=(m_top-m_low)/(N-1);   % rozdělení na N pásem v mel stupnici
    xm=m_low:mdiv:m_top;  % střední frekvence každého pásma
    xf=mel2f(xm); % střední frekvence převedené zpět na Hertzovou stupnici
    xq = floor((length(frame)/2 + 1)*xf/(Fs/2));
    S=fft(frame); % výpočet FFT pro signál ve frame
    S=abs(2*(S.*S)/length(S)); % výpočet energie (kvadrátu) každé složky
    S=S(1:length(S)/2); % výběr první poloviny hodnot
    F=[1:length(S)]*(Fs/2)/length(S);  % vydělení počtem vzorků
    x1=zeros(1,N);
    for xi=1:N    % cyklus pro všechna pásma
        band=spread_mel(xf,xi,length(S),Fs/2); % trojúhelníková funkce
        x1(xi)=sum(band.*S');   % výpočet energie v pásmu
    end
    x=log(x1);   % výpočet logaritmu energie pro vsechna pásma
    mel_fbank = x;  % mel-spektralni příznaky spočítány
    cep_coeff=zeros(1,M);
    for xc=1:M   % disktrétní kosinová transformace
        cep_coeff(xc)=sqrt(2/N)*sum(x.*cos(pi*xc*([1:N]-0.5)/N));
    end
end
```

Jak zrychlit klasifikátor

Obecné poznámky:

1) V Matlabu zkuste **vyloučit smyčky** a místo nich použít maticové a vektorové operace.

2) Zjistěte, které části kódu jsou **nejčastěji opakované,** a zaměřte se na jejich **optimalizaci**.

3) **Přesuňte** co nejvíce výpočtů **z vnitřních smyček** na místa, kde budou prováděny pouze jednou, tj. do vnějších smyček nebo do inicializace

4) Hledejte takové části v kódu, které **mohou být vynechány, aniž by se změnila činnost programu a jeho výsledky**
   např. u *Itakurových podmínek* můžete vynechat výpočty pro body, ležící *mimo globální podmínky*

5) Při praktické realizaci použijte jiné jazyky:
   Jazyk C umožní mnohem rychlejší kód, lze ale použít i **VB** či jiné jazyky, které umožní **kompilaci do *.exe kódu**

Výpočet lokální vzdálenosti

U metody DTW je to nejčastěji prováděná operace

Příklad: má-li testované slovo 100 framů, refence 100 framů, musí se provést 100x100 výpočtů vzdáleností.

Ukažme si několik způsobů výpočtu z hlediska rychlosti:

```
word(f,:), ref(f,:)  je příznakový vektor framu f slova či reference
```

1) Klasická smyčka realizující Euklidovskou vzdálenost (níže) je příliš pomalá

```
dist=0; for p=1:počet_priznaku; dist=dist+(word(f,p)-ref(f,p))^2; end;
```

2) Rychlejší už je maticová operace

```
dist = sum((word(f,:) - ref(f,:))^2)
```

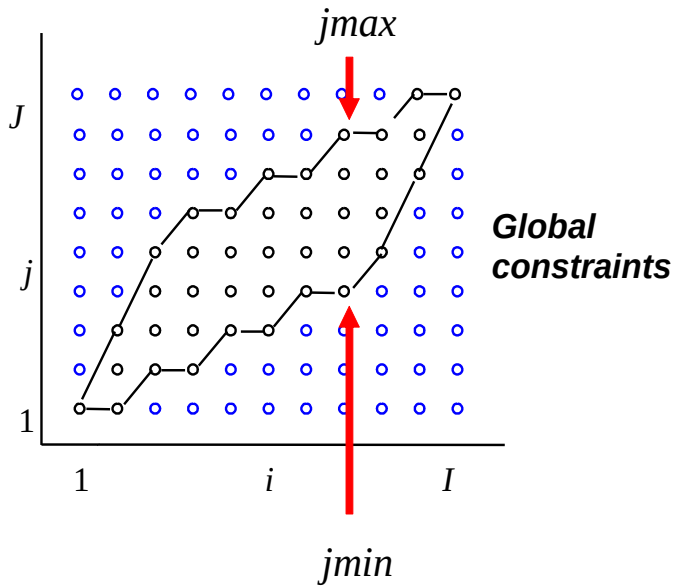
3) Ještě rychlejší je ale speciální funkce norm určená přímo pro Eukl. vzdál.

```
dist = norm(word(f,:) - ref(f,:))
```

Použití globálních podmínek

U Itakurových podmínek

mnoho bodů v mřížce může být vynecháno z výpočtu, aniž by to mělo vliv na výsledek



Typicky **více než 2/3 bodů** leží mimo hranici danou globálními podmínkami

Příklad rutiny ve VB, která počítá ***jmin*** a ***jmax*** *(v programu označené jako Mi a Mx)*

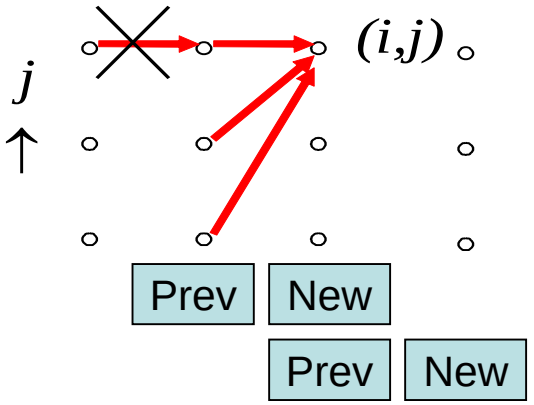
```
Sub MinMax(Mi As Integer, Mx As Integer, i As Integer, _
           TestLength As Integer, RefLength As Integer)
   Dim K As Integer, K1 As Integer
   ' For Itakura constraints
   ' This routine computes mimimum (Mi) and maximum (Mx) for each i
   K = (i + 1) \ 2     ' symbol \ means integer division
   K1 = RefLength - (TestLength - i) * 2
   If K < K1 Then K = K1
   Mi = K
   K = 2 * i - 1
   K1 = RefLength - (TestLength - i) \ 2
   If K > K1 Then K = K1
   Mx = K

End Sub
```

Optimalizace hlavní rutiny (1)

Potřebujeme kompletní pole A(i,j) a B(i,j)?

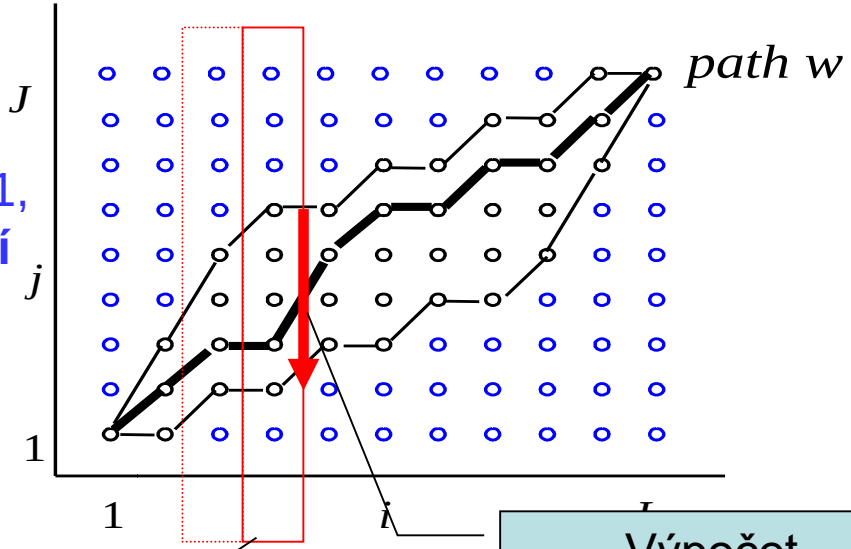
Odpověď: obvykle ne

- pole B je nutné **jen** kvůli backtrackingu
- z pole A(i,j) potřebujeme pouze **předchozí** a **současný sloupec** a dále **informaci** zda předchozí nejlepší cesta byla **horizontální**



A možná je ještě větší úspora:

Místo 2 sloupců můžeme použít pouze 1, v něm musíme **přepisovat předchozí** hodnoty **novými hodnotami**. Pozor, musíme to pak dělat ve směru **zhora dolů.**



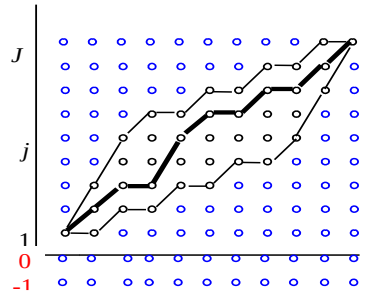
Sloupec hodnot, který se „posouvá“ doprava

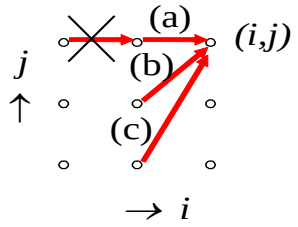
Výpočet zhora dolů

Optimalizace hlavní rutiny(2)

Ukázkový kód ve VB:

```
Function dtwItakura(TestLength As Integer, RefLength As Integer) As Single
   Static accDist(1 To MAX_FRAMES)  As Single 'one column of acummulated distances
   Static wasHorizontal(1 To MAX_FRAMES)  As Integer 'carries info if previous path was horizonta
   Dim i As Integer, j As Integer, Mi As Integer, Mx As Integer, temp As Single

   If (TestLength > 2 * RefLength) Or (RefLength > 2 * TestLength - 1) Then 'Test for lengths
      dtwItakura = BIG: Exit Function  'if lengths differ too much, distance cannot be computed
   End If

   For j = -1 To RefLength 'Initial values (for i = 1)
      jj = j + 2      ' jj is temporal variable to avoid negative indeces
      accDist(jj) = BIG: wasHorizontal(jj) = 0
   Next j
   accDist(1 + 2) = locdist(1, 1)

   For i = 2 To TestLength ' Loop for i
      Call MinMax(Mi, Mx, i, TestLength, RefLength, 0) 'Find min and max values for j
      For j = Mx To Mi Step -1 'Go from top to down, so you can replace old values by new
         jj = j + 2 'temporary variable
         temp = accDist(jj - 2)
         If temp > accDist(jj - 1) Then 'find which is better: steep or diagonal
            temp = accDist(jj - 1)
         End If
         If (wasHorizontal(jj)) = 0 And (accDist(jj) < temp) Then 'find if horizontal is better
            wasHorizontal(jj) = 1: temp = accDist(jj)
         Else
            wasHorizontal(jj) = 0
         End If
         accDist(jj) = temp + locDist(i, j) ' add local distance
      Next j
      accDist(Mi + 2 - 1) = BIG ' set the value below constraints BIG
   Next i

   dtwItakura = accDist(RefLength + 2) 'Now we have the global distance
End Function
```





Samostatná úloha

Sestavte si DTW rozpoznávač pracující s příznaky Spec, Fbank a MFCC a experimentálně ho vyhodnoťte na datech

1) Na základě dnešní přednášky si připravte a otestujte modul pro výpočet příznaků Fbank a MFCC (programy si stáhněte z elearningu)
2) Dále se snažte co nejvíce zrychlit DTW modul. Čas výpočtu měřte pomocí funkcí tic toc.
3) Použijte stejná data jako minule (2021, 2022, 2023) S nimi proveďte experimenty, které porovnají 3 typy příznaků: Spec16, Fbank26, a MFCC12
4) Celkem provedete tedy 3 experimenty, přičemž každý se bude skládat z rozpoznání testovacích dat od jednotlivých osob a výpočtu průměrné úspěšnosti přes všechny tyto osoby. Program také vypíše dobu výpočtu pro každý experiment.

Samostatná úloha (pokrač.)

5) U každé osoby použijte sadu 01 jako trénovací (tj. jako reference) a ostatní sady jako testovací (tj. jako dosud). Spočítejte úspěšnost pro každou osobu a následně průměrnou hodnotu přes všechny osoby.
6) Nejdéle do pondělí 12.00 mi pošlete váš program (pracující se souborem FileList.txt) a dále průměrné hodnoty úspěšnosti ze všech 3 experimentů s danými typy příznaků a rovněž dobu trvání každého kompletního experimentu.
7) Výsledek uveďte v mailu – 3 řádky typu:
   Příznaková sada: …. Úspěšnost = …. Trvání experimentu
8) Váš program spustím s vlastním FileListem, v němž budou cesty ke stejným souborům na mém počítači, a měl by fungovat.
9) Příště zveřejním nejlepší dosažené výsledky.