Mel Frequency Cepstral Coefficient (MFCC) tutorial

The first step in any automatic speech recognition system is to extract features i.e. identify the components of the audio signal that are good for identifying the linguistic content and discarding all the other stuff which carries information like background noise, emotion etc.

The main point to understand about speech is that the sounds generated by a human are filtered by the shape of the vocal tract including tongue, teeth etc. This shape determines what sound comes out. If we can determine the shape accurately, this should give us an accurate representation of the phoneme being produced. The shape of the vocal tract manifests itself in the envelope of the short time power spectrum, and the job of MFCCs is to accurately represent this envelope. This page will provide a short tutorial on MFCCs.

Mel Frequency Cepstral Coefficents (MFCCs) are a feature widely used in automatic speech and speaker recognition. They were introduced by Davis and Mermelstein in the 1980's, and have been state-of-the-art ever since. Prior to the introduction of MFCCs, Linear Prediction Coefficients (LPCs) and Linear Prediction Cepstral Coefficients (LPCCs) (click here for a tutorial on cepstrum and LPCCs) and were the main feature type for automatic speech recognition (ASR), especially with HMM classifiers. This page will go over the main aspects of MFCCs, why they make a good feature for ASR, and how to implement them.

Steps at a Glance

We will give a high level intro to the implementation steps, then go in depth why we do the things we do. Towards the end we will go into a more detailed description of how to calculate MFCCs.

1. Frame the signal into short frames.
2. For each frame calculate the periodogram estimate of the power spectrum.
3. Apply the mel filterbank to the power spectra, sum the energy in each filter.
4. Take the logarithm of all filterbank energies.
5. Take the DCT of the log filterbank energies.
6. Keep DCT coefficients 2-13, discard the rest.

There are a few more things commonly done, sometimes the frame energy is appended to each feature vector. Delta and Delta-Delta features are usually also appended. Liftering is also commonly applied to the final features.

Why do we do these things?

We will now go a little more slowly through the steps and explain why each of the steps is necessary.

An audio signal is constantly changing, so to simplify things we assume that on short time scales the audio signal doesn't change much (when we say it doesn't change, we mean statistically i.e. statistically stationary, obviously the samples are constantly changing on even short time scales). This is why we frame the signal into 20-40ms frames. If the frame is much shorter we don't have enough samples to get a reliable spectral estimate, if it is longer the signal changes too much throughout the frame.

The next step is to calculate the power spectrum of each frame. This is motivated by the human cochlea (an organ in the ear) which vibrates at different spots depending on the frequency of the incoming sounds. Depending on the location in the cochlea that vibrates (which wobbles small hairs), different nerves fire informing the brain that certain frequencies are present. Our periodogram estimate performs a similar job for us, identifying which frequencies are present in the frame.

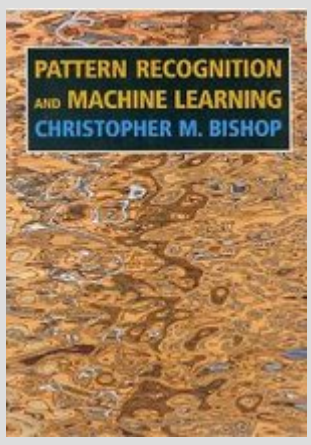
The periodogram spectral estimate still contains a lot of information not required for Automatic Speech Recognition (ASR). In particular the cochlea can not discern the difference between two closely spaced frequencies. This effect becomes more pronounced as the frequencies increase. For this reason we take clumps of periodogram bins and sum them up to get an idea of how much energy exists in various frequency regions. This is performed by our Mel filterbank: the first filter is very narrow and gives an indication of how much energy exists near 0 Hertz. As the frequencies get higher our filters get wider as we become less concerned about variations. We are only interested in roughly how much energy occurs at each spot. The Mel scale tells us exactly how to space our filterbanks and how wide to make them. See below for how to calculate the spacing.

Once we have the filterbank energies, we take the logarithm of them. This is also motivated by human hearing: we don't hear loudness on a linear scale. Generally to double the percieved volume of a sound we need to put 8 times as much energy into it. This means that large variations in energy may not sound all that different if the sound is loud to begin with. This compression operation makes our features match more closely what humans actually hear. Why the logarithm and not a cube root? The logarithm allows us to use cepstral mean subtraction, which is a channel normalisation technique.

Contents

Further reading

We recommend these books if you're interested in finding out more.

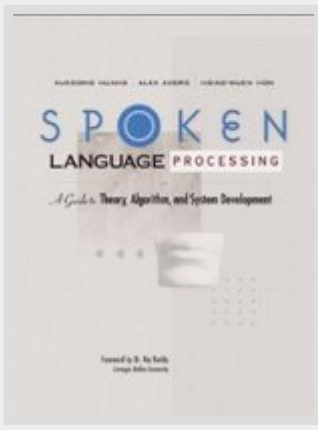


Pattern Recognition and Machine Learning

ASIN/ISBN: 978-0387310732

"The best machine learning book around"

Buy from Amazon.com



Spoken Language Processing: A Guide to Theory, Algorithm and System Development

ASIN/ISBN: 978-0130226167

"A good overview of speech processing algorithms and techniques"

Buy from Amazon.com

The final step is to compute the DCT of the log filterbank energies. There are 2 main reasons this is performed. Because our filterbanks are all overlapping, the filterbank energies are quite correlated with each other. The DCT decorrelates the energies which means diagonal covariance matrices can be used to model the features in e.g. a HMM classifier. But notice that only 12 of the 26 DCT coefficients are kept. This is because the higher DCT coefficients represent fast changes in the filterbank energies and it turns out that these fast changes actually degrade ASR performance, so we get a small improvement by dropping them.

What is the Mel scale?

The Mel scale relates perceived frequency, or pitch, of a pure tone to its actual measured frequency. Humans are much better at discerning small changes in pitch at low frequencies than they are at high frequencies. Incorporating this scale makes our features match more closely what humans hear.

The formula for converting from frequency to Mel scale is:

$$M(f) = 1125 \ln(1 + f/700) \qquad (1)$$

To go from Mels back to frequency:

$$M^{-1}(m) = 700(\exp(m/1125) - 1) \qquad (2)$$

Implementation steps

We start with a speech signal, we'll assume sampled at 16kHz.

1. Frame the signal into 20-40 ms frames. 25ms is standard. This means the frame length for a 16kHz signal is 0.025*16000 = 400 samples. Frame step is usually something like 10ms (160 samples), which allows some overlap to the frames. The first 400 sample frame starts at sample 0, the next 400 sample frame starts at sample 160 etc. until the end of the speech file is reached. If the speech file does not divide into an even number of frames, pad it with zeros so that it does.

The next steps are applied to every single frame, one set of 12 MFCC coefficients is extracted for each frame. A short aside on notation: we call our time domain signal $s(n)$. Once it is framed we have $s_i(n)$ where n ranges over 1-400 (if our frames are 400 samples) and i ranges over the number of frames. When we calculate the complex DFT, we get $S_i(k)$ - where the i denotes the frame number corresponding to the time-domain frame. $P_i(k)$ is then the power spectrum of frame i.

2. To take the Discrete Fourier Transform of the frame, perform the following:

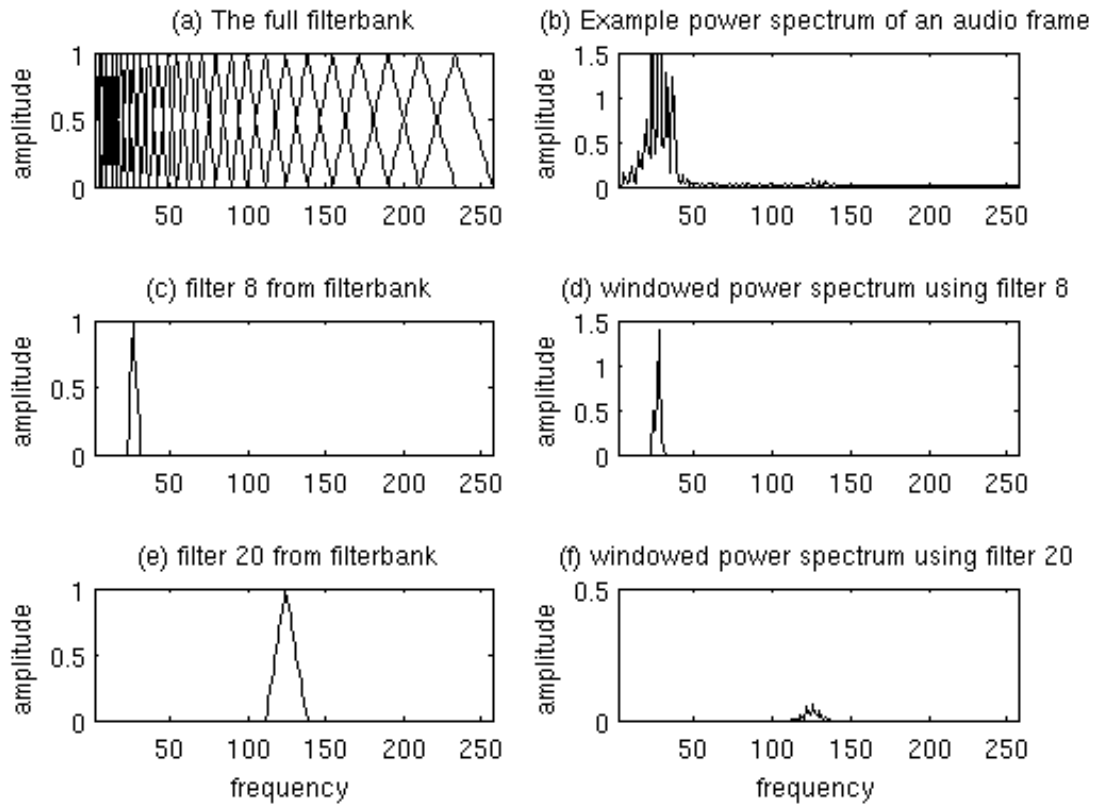
$$S_i(k) = \sum_{n=1}^{N} s_i(n)h(n)e^{-j2\pi kn/N} \qquad 1 \leq k \leq K$$

where $h(n)$ is an N sample long analysis window (e.g. hamming window), and K is the length of the DFT. The periodogram-based power spectral estimate for the speech frame $s_i(n)$ is given by:

$$P_i(k) = \frac{1}{N}|S_i(k)|^2$$

This is called the Periodogram estimate of the power spectrum. We take the absolute value of the complex fourier transform, and square the result. We would generally perform a 512 point FFT and keep only the first 257 coefficents.

3. Compute the Mel-spaced filterbank. This is a set of 20-40 (26 is standard) triangular filters that we apply to the periodogram power spectral estimate from step 2. Our filterbank comes in the form of 26 vectors of length 257 (assuming the FFT settings fom step 2). Each vector is mostly zeros, but is non-zero for a certain section of the spectrum. To calculate filterbank energies we multiply each filterbank with the power spectrum, then add up the coefficents. Once this is performed we are left with 26 numbers that give us an indication of how much energy was in each filterbank. For a detailed explanation of how to calculate the filterbanks see below. Here is a plot to hopefully clear things up:



Plot of Mel Filterbank and windowed power spectrum

4. Take the log of each of the 26 energies from step 3. This leaves us with 26 log filterbank energies.

5. Take the Discrete Cosine Transform (DCT) of the 26 log filterbank energies to give 26 cepstral coefficents. For ASR, only the lower 12-13 of the 26 coefficients are kept.

The resulting features (12 numbers for each frame) are called Mel Frequency Cepstral Coefficients.

Computing the Mel filterbank

In this section the example will use 10 filterbanks because it is easier to display, in reality you would use 26-40 filterbanks.

To get the filterbanks shown in figure 1(a) we first have to choose a lower and upper frequency. Good values are 300Hz for the lower and 8000Hz for the upper frequency. Of course if the speech is sampled at 8000Hz our upper frequency is limited to 4000Hz. Then follow these steps:

1. Using equation 1, convert the upper and lower frequencies to Mels. In our case 300Hz is 401.25 Mels and 8000Hz is 2834.99 Mels.
2. For this example we will do 10 filterbanks, for which we need 12 points. This means we need 10 additional points spaced linearly between 401.25 and 2834.99. This comes out to:

   ```
   m(i) = 401.25, 622.50, 843.75, 1065.00, 1286.25, 1507.50, 1728.74,
          1949.99, 2171.24, 2392.49, 2613.74, 2834.99
   ```

3. Now use equation 2 to convert these back to Hertz:

   ```
   h(i) = 300, 517.33, 781.90, 1103.97, 1496.04, 1973.32, 2554.33,
          3261.62, 4122.63, 5170.76, 6446.70, 8000
   ```

   Notice that our start- and end-points are at the frequencies we wanted.
4. We don't have the frequency resolution required to put filters at the exact points calculated above, so we need to round those frequencies to the nearest FFT bin. This process does not affect the accuracy of the features. To convert the frequncies to fft bin numbers we need to know the FFT size and the sample rate,

   ```
   f(i) = floor((nfft+1)*h(i)/samplerate)
   ```

   This results in the following sequence:

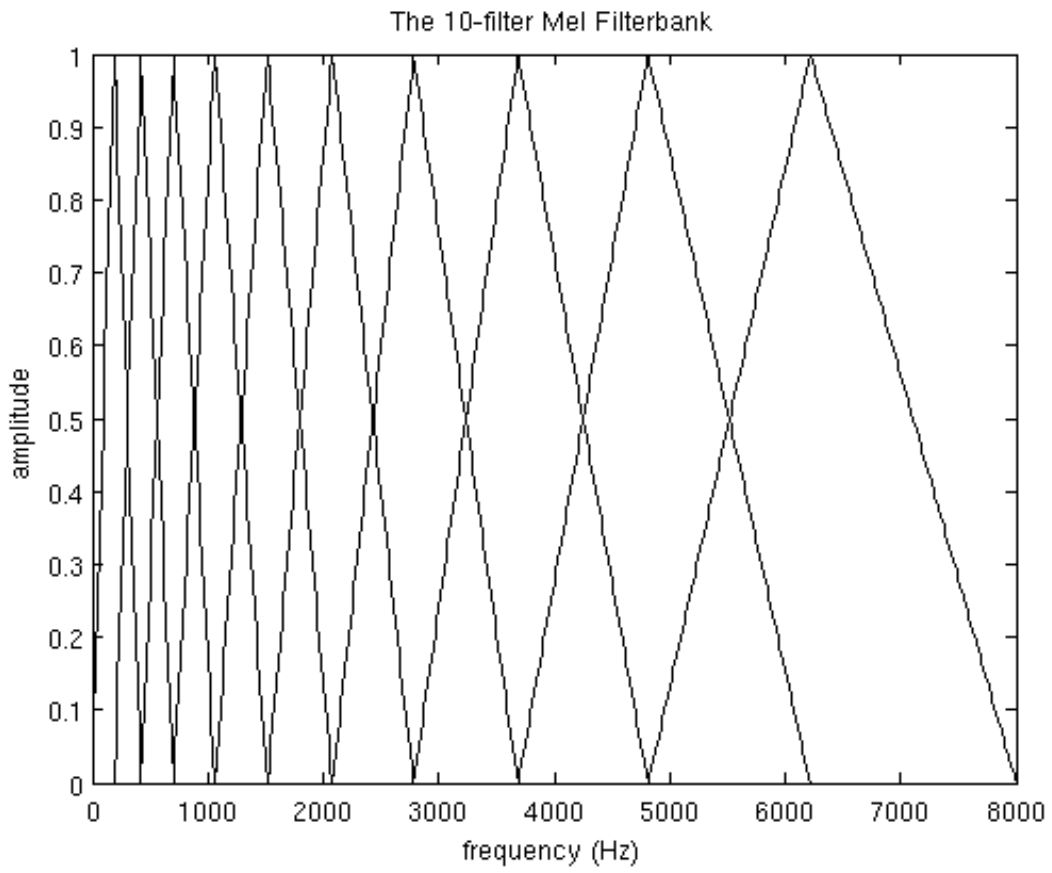
   ```
   f(i) =  9, 16,  25,   35,   47,   63,   81,  104,  132, 165,  206,  256
   ```

   We can see that the final filterbank finishes at bin 256, which corresponds to 8kHz with a 512 point FFT size.
5. Now we create our filterbanks. The first filterbank will start at the first point, reach its peak at the second point, then return to zero at the 3rd point. The second filterbank will start at the 2nd point, reach its max at the 3rd, then be zero at the 4th etc. A formula for calculating these is as follows:

$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \dfrac{k - f(m-1)}{f(m) - f(m-1)} & f(m-1) \le k \le f(m) \\ \dfrac{f(m+1) - k}{f(m+1) - f(m)} & f(m) \le k \le f(m+1) \\ 0 & k > f(m+1) \end{cases}$$

   where M is the number of filters we want, and $f()$ is the list of M+2 Mel-spaced frequencies.

The final plot of all 10 filters overlayed on each other is:



A Mel-filterbank containing 10 filters. This filterbank starts at 0Hz and ends at 8000Hz. This is a guide only, the worked example above starts at 300Hz.

Deltas and Delta-Deltas

Also known as differential and acceleration coefficients. The MFCC feature vector describes only the power spectral envelope of a single frame, but it seems like speech would also have information in the dynamics i.e. what are the trajectories of the MFCC coefficients over time. It turns out that calculating the MFCC trajectories and appending them to the original feature vector increases ASR performance by quite a bit (if we have 12 MFCC coefficients, we would also get 12 delta coefficients, which would combine to give a feature vector of length 24).

To calculate the delta coefficients, the following formula is used:

$$d_t = \frac{\sum_{n=1}^{N} n(c_{t+n} - c_{t-n})}{2\sum_{n=1}^{N} n^2}$$

where d_t is a delta coefficient, from frame t computed in terms of the static coefficients c_{t+N} to c_{t-N}. A typical value for N is 2. Delta-Delta (Acceleration) coefficients are calculated in the same way, but they are calculated from the deltas, not the static coefficients.

Implementations

I have implemented MFCCs in python, available here. Use the 'Download ZIP' button on the right hand side of the page to get the code. Documentation can be found at readthedocs. If you have any troubles or queries about the code, you can leave a comment at the bottom of this page.

There is a good MATLAB implementation of MFCCs over here.

References

Davis, S. Mermelstein, P. (1980) *Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences*. In IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 28 No. 4, pp. 357–366

X. Huang, A. Acero, and H. Hon. *Spoken Language Processing: A guide to theory, algorithm, and system development*. Prentice Hall, 2001.

Related pages on this site:

- A tutorial on LPCCs and Cepstrum
- Hidden Markov Model (HMM) tutorial
- Gaussian Mixture Models (GMMs) and the EM Algorithm
- An Intuitive Guide to the Discrete Fourier Transform

Name

Rosewater • 9 years ago

You need to write a book. You are a master at explaining these concepts. I like the fact that you never assume prerequisite knowledge is "obvious," and explain every detail. Thank you so much.

57 | 1 • Reply • Share ›

jameslyons Mod ➚ Rosewater • 9 years ago

thanks for the compliment, I am glad you found it useful!

12 | 1 • Reply • Share ›

Albert Tayong ➚ jameslyons • 5 years ago

yeah , but what is value of the floor for each?

| • Reply • Share ›

jameslyons Mod ➚ Albert Tayong • 5 years ago

Floor is a function https://en.m.wikipedia.org/...

| • Reply • Share ›

Albert Tayong ➚ jameslyons • 5 years ago

Hi james

Can u please explain Si(k)

How to find si(n) anf h(n)

8 | 1 • Reply • Share ›

Albert Tayong ➚ jameslyons • 5 years ago

Ok..thanks i got it

| • Reply • Share ›

Juan Fonseca ➚ jameslyons • 5 years ago • edited

Seriously James, put this in paper or a book. My supervisor won't let me cite anything if it is not indexed. I have found this page extremely useful, thanks!

| • Reply • Share ›

Marcio ➚ Juan Fonseca • 5 years ago

The same goes for me. This is better explained than a lot of papers that I found. There's even a paper that I found that cites you.

| • Reply • Share ›

jameslyons Mod ➚ Juan Fonseca • 5 years ago

That is a good idea, I might try to find a low level publication somewhere and submit it. I'm glad you like i!

| • Reply • Share ›

arracso ➚ jameslyons • 2 years ago

did you put it on a paper already??

| • Reply • Share ›

Benspy001 ➚ arracso • a year ago

4 years but yet useful. Did he ever make a paper?

| • Reply • Share ›

Syed Zubair Zahid ➚ Rosewater • 3 years ago

HE is amazing

1 | • Reply • Share ›

Albert Tayong ➚ Rosewater • 5 years ago

HOW DID HE GET 9 IN THE FIRST f(I) and 16 and so on and so forth.

the formula is given, but there is no example how to get 9, 16, etc.HOW DID HE GET 9 IN THE FIRST f(I) and 16 and so on and so forth.

the formula is given, but there is no example how to get 9, 16, etc.

| • Reply • Share ›

jameslyons Mod ➚ Albert Tayong • 5 years ago

Just put the numbers into the formula? Samplerate is 16k, nfft is 512, h (i) is listed just above, just put the numbers in a calculator.

1 | • Reply • Share ›

D.Horse • 8 years ago

Thanks, great article!

28 ^ | v • Reply • Share ›

keoki1 • 7 years ago

@jameslyons , I computed the MFCC coefficients and I currently have a 2D array Number of Frames * Coefficients (MFCC and Deltas) (260 x 36). In order to check if 2 speakers are the same, i am computing the distance using DTW . DTW(coefficients1, coefficients2) , the one with the lowest distance is most probably the speaker. Before passing the coefficients to the DTW, I normalized all 260 x 36 coefficients between -1 and +1. I am testing against a database of 83 sounds (5 sounds / speaker) but I am always obtaining the wrong results. Is there something that I am missing? Many thanks!

28 ^ | v 1 • Reply • Share ›

Majid • 9 years ago

Hi

I'm So sorry for this request :
Someone can help me to explain step 4 of computing the Mel Filter bank with actual data. just 2,3 step.

My English is not very good.

Thank you

35 ^ | v 2 • Reply • Share ›

teja polisetty • 4 years ago

those 26 coefficents of MFCC which we are removing and finally keeping last 12 coefficents right. Can you please tell like what are those coeficents actually, and how it varies for different emotion.
Thanks in advance :)

19 ^ | v 1 • Reply • Share ›

SSH • 9 years ago

I just want to say THANK YOU! This is the first "easy to understand" explanation of MFCCs that I have come across. This article answers perfectly the two aspects of any concept that I seek i.e. the WHY and HOW of things. I plan to explore this website to the full extent and grab all the goodies it has to offer. :)

10 ^ | v • Reply • Share ›

talha • 9 years ago

hi It is a very nice tutorial.I am complete beginer in speech processing field.i searched for some voice feature extraction softwares. (I found one) i have a sound sample, by applying window length 0.015 and time step 0.005, i have extracted 12 MFCC features for 171 frames directly from sound sample by using software tool called PRAAT. Now I have all 12 MFCC coefficients for each frame.. my question is that now i want to process them further making there 39 dimensional matrix by adding energy feature and delta-delta features and apply dtw. I dont know how to deal with coefficients and how to make delta-delta coefficients. I am having trouble for using above farmula . can you please guide me step by step i am complete beginener and in a lot of trouble..

16 ^ | v 3 • Reply • Share ›

Hunaa ➚ talha • 4 years ago

How you resolved this issue?

^ | v • Reply • Share ›

Zahidul islam • 3 years ago

m(i) = 401.25, 622.50, 843.75, 1065.00, 1286.25, 1507.50, 1728.74,
1949.99, 2171.24, 2392.49, 2613.74, 2834.99
where i found this data as i saw the difference is 221.25 but why we assume it.My another query is when we perform delta deltas .please sir explain me.
thanks advance

5 ^ | v • Reply • Share ›

Anuraj ➚ Zahidul islam • a year ago

hear we need 10 banks. so 2834.99-401.25 = 2,433.74, then we need to define 2,433.74 /11 = 221.249...

Thats why we get 221.25

^ | v • Reply • Share ›

Phong Đỗ • 6 years ago

Hello, Thanks for your tutorial.
From my understanding, we divide the signal into small frames (20ms in my case, 10ms

shift) and compute MFCC for each frame independently, not interact with other frames. However, when I remove a piece of signal lasting 20ms in time domain and compute MFCC, I expect that only frames overlapping removing-signal portion will change. Unfortunately, other MFCC frames after removing-signal part changed too.
Is there any misunderstanding from me?
Thank you

3 • Reply • Share ›

Ionna • 8 years ago

About the DCT: I have read that it makes the transformation back to time domain. Why that? You are saying that it is used for decorrelation. What is its purpose after all? Could you explain more or put a link? Thanks a lot!

4 • 1 • Reply • Share ›

Robert → Ionna • 6 years ago

I found a great answer about the DCT part.

https://tspace.library.utor...

Page 52

The 26 bins are called MFSC (spectral coefficients).

The 13 values after the DCT are the MFCC.

The thesis I linked explains why the DCT is made to increase results of Gaussian models and MFSC are better for Neural Network based models.

• 1 • Reply • Share ›

riti • 4 years ago

Thanks a lot.Your explanation is so simple and lucid..!!

2 • Reply • Share ›

Siavash • 9 years ago

Hi,
i create 10 filter bank
Now,how do I compress these down to 256 elements?

2 • Reply • Share ›

Miloš Pušica • 4 years ago

Hi, thanks for the great article...it's really well explained! I noticed just one detail that is not clear to me.
Why did you take 300Hz as your lower boundary for the first filterbank? I can see from the picture of filterbanks that the first filterbank starts at 0Hz, which is reasonable to me. Putting the lower boundary at 300Hz would discard all the information from spectrum below 300Hz.

1 • Reply • Share ›

Nikas • 6 years ago

I do not understand this part "This is called the Periodogram estimate of the power spectrum. We take the absolute value of the complex fourier transform, and square the result. We would generally perform a 512 point FFT and keep only the first 257 coefficents." where do we get those 512 point and 257 coefficients?

1 • Reply • Share ›

Bart → Nikas • 6 years ago

From FFT. FFT will result in a set of coefficients, then just keep however many you need.

• Reply • Share ›

Nikas → Bart • 6 years ago

So basically after Periodogram estimation of the power spectrum I need to do FFT? And are there any options to decide how many coefficients I need to choose?

• Reply • Share ›

Abhijeet Singh • 7 years ago • edited

I loved this tutorial explaining MFCC.
In your python implementation of MFCC, Do the mfcc feature represents the energy of the frame?
I have read in literature n books that the first coeff of MFCC represents the energy of frame.
I extracted MFCCs from your implemented code but it does not seem to give the energy or log energy of frame.
So, i wanted to know how do your implementation (python) take this in consideration.?
Or i have to append the energy of frame as a feature in my code to the features extracted from your implementation???

Thank you.

1 • Reply • Share ›

jameslyons Mod ➚ Abhijeet Singh • 7 years ago

see line 35 of base.py (https://github.com/jameslyo...) It is where the first MFCC coefficient is replaced with the log of the frame energy.

• Reply • Share ›

Abhijeet Singh ➚ jameslyons • 7 years ago

Thank you...i didn't see it initially. thanks...

• Reply • Share ›

jameslyons Mod ➚ Abhijeet Singh • 7 years ago

no problem :) happy to help.

• Reply • Share ›

Sami Liedes • 7 years ago

If you want to take the logs of the bin energies, shouldn't you rather do something like log(1+energy)? I'm toying with your Python code, and for near-silent portions of signal logfbank() returns roughly -36, which wracks the DCT.

1 • Reply • Share ›

jameslyons Mod ➚ Sami Liedes • 7 years ago

That is a problem with log, other features like PLP use a cube-root function for compressing the energies instead of log, which has much nicer behaviour for small numbers. Though if you use cube root you can't do cepstral mean subtraction any more. In the end it is not critical exactly how it is done, as long as the recognition results are good. If you get better results with log(1+x), you should use it.

1 • Reply • Share ›

Dikshit Nagaraj • 7 years ago

Hi,
sir i found the 26 MFCC+delta-delta for each frame in an audio signal..[In MatLab]
i have 8 signals n i found the same for all the signals..

Now in my application if the speaker says one of those 8 words it should be able to identify that the speaker said one of those 8 words.

My problem is how can i do tht ?Im stuck ..what am i supposed to do with those features...

some one plz help.

Thank you..

1 • Reply • Share ›

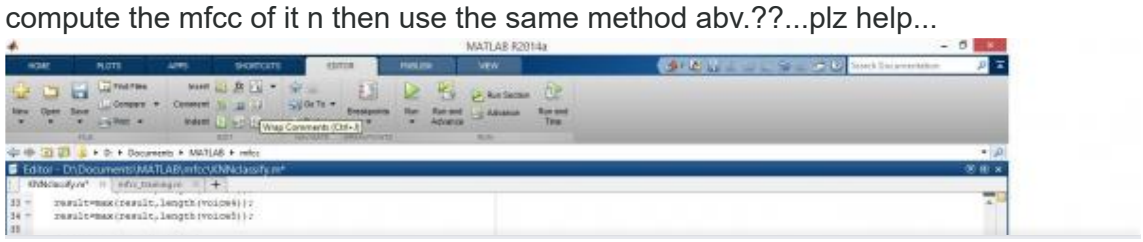
Dikshit Nagaraj ➚ Dikshit Nagaraj • 7 years ago

I tried converting each signal matrix [rows x 26] into one row matrix [1 x columns]..
so i for 8 signals i was left with [8 x columns]....thn i used
knnclassify(myvoiceinput,training,group).....
where
Training=tht [8 x column] matrix

group=[8 x 1]matrix indicating the lables for each signal...

but im getting the wrong ans...
Plz check the images n tell if got the correct plots for mfcc+delta..n also check if my code is correct..

im i doing anything wrong..should i repeat take more recordings for the same word n compute the mfcc of it n then use the same method abv.??...plz help...



see more

31 • Reply • Share ›

hafida ➚ Dikshit Nagaraj • 6 years ago

Hello,
i need a code matlab for MFCC extraction for speech wav.
can you give it to me.
please I need your help
thanks in advance

• Reply • Share ›

nir • 8 years ago

but how will we get only 26 values by using 10 filters ???? or the 26 coefficient is independent from no. of filter??

1 ^ | v • Reply • Share ›

jameslyons Mod ➔ nir • 8 years ago

yeah, so I guess I should have made that clearer, sorry. The picture shows 10 filters because 26 was too crowded, that is the only reason. Normally you want to use 26 filters. Ignore the 10.

^ | v • Reply • Share ›

SUNNY • 8 years ago

In Compute mel filter banks can you explain what these variables are???

$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \dfrac{k - f(m-1)}{f(m) - f(m-1)} & f(m-1) \le k \le f(m) \\ \dfrac{f(m+1) - k}{f(m+1) - f(m)} & f(m) \le k \le f(m+1) \\ 0 & k > f(m+1) \end{cases}$$

1 ^ | v • Reply • Share ›

shruti • 8 years ago

never thought that i can understand this topic.....thanks a lot....our professors need to learn from u how to teach

1 ^ | v • Reply • Share ›

loana • 9 years ago

This is the best detailed explanation I found of how to compute the MFCC. Thank you! (I'm sorry I didn't find it days ago, it would have spared me a lot of time...)

1 ^ | v • Reply • Share ›

monica ➔ loana • 7 years ago

can u pl explain abt delta and delta-delta coeff

^ | v • Reply • Share ›

akashreddyk • 6 months ago

f(i) = floor((nfft+1)*h(i)/samplerate)
is nfft a command ?

^ | v • Reply • Share ›

jameslyons Mod ➔ akashreddyk • 6 months ago

No, the number of fft bins you want to use e.g. 512 or 1024

^ | v • Reply • Share ›

rh c • 7 months ago

It's my first time learn the extract MFCC from speech. thank you i will be better to know it

^ | v • Reply • Share ›

Load more comments

Subscribe Add Disqus to your siteAdd DisqusAdd Do Not Sell My Data

hzmdohwfzhh oh fju monofa ch jfz vohwzh uj monz, ou oh chbofa juwzyh uj monz ch jfz vohwzh uj monz
– jhqcy vomtz

Questions/Feedback

Notice a problem? We'd like to fix it!
Leave a comment on the page and we'll take a look.