MSPR 10: Gaussian Mixture Models and Kernel Smoothing

Dr. Hendrik Purwins

AAU CPH

November 2, 2015

- Chris Bishop: Pattern Recognition and Machine Learning pp. 430-439 (GMM), pp. 122-124 (Kernel Smoothing)
- Andrew Moore's tutorial: http://www.autonlab.org/tutorials/gmm14.pdf

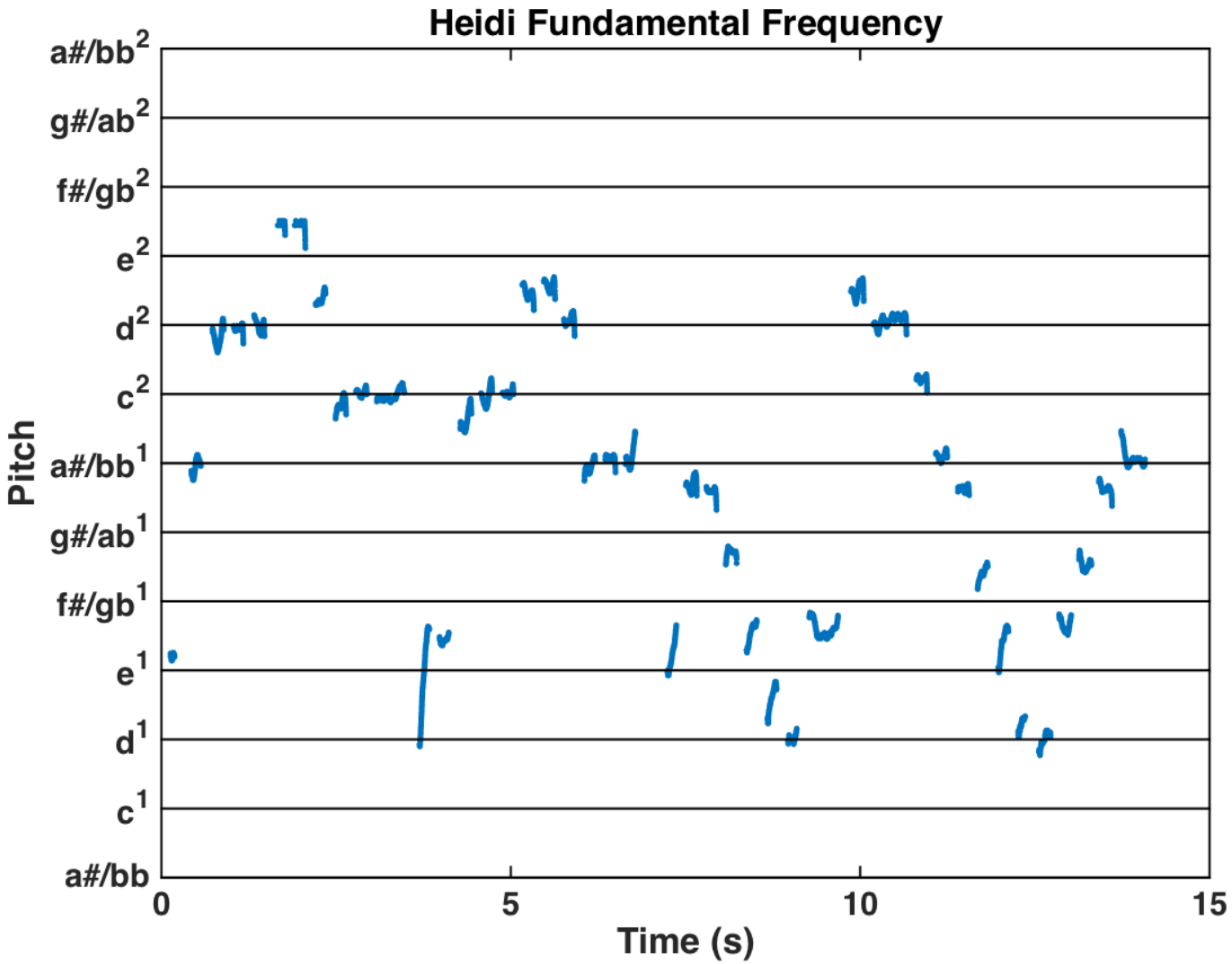
Outline

Aliens in the Alps I

- Aliens from Sirius have been stranded in the Alps.
- These aliens have never heard Western music, but have the same auditory system and the same statistic and sound analysis skills as you.
- They hear Heidi singing (download `heidi.wav` from moodle).
- How can they make sense of what they hear?
- Let us listen just to the fundamental frequency. (download `heidif0.wav` from moodle).
- How many different notes are there?
- Then they plot the frequency content of what they hear!

Aliens in the Alps II

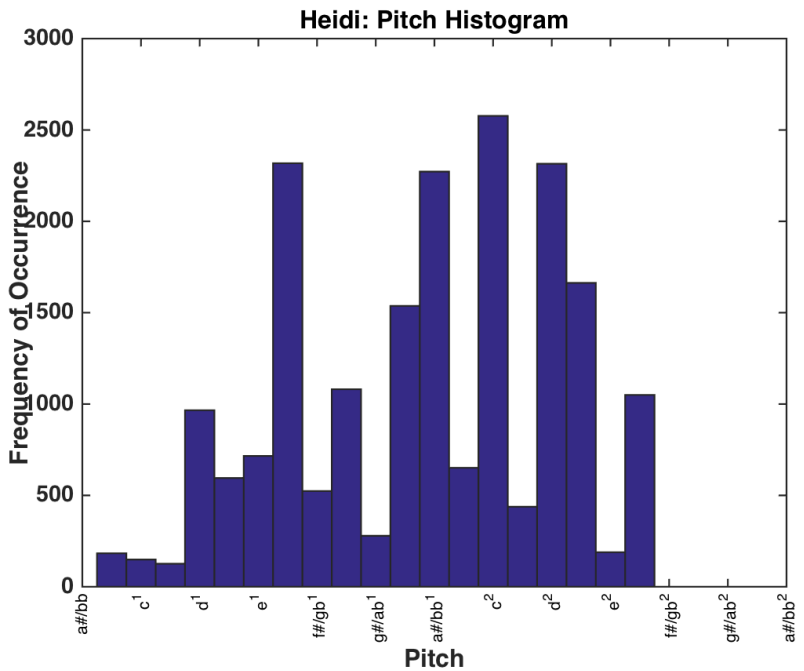
```
load heidi_analysis.mat
labs={'a#/bb','c^1','d^1','e^1','f#/gb^1','g#/ab^1','a#/
    bb^1','c^2','d^2','e^2','f#/gb^2','g#/ab^2','a#/bb^2
    '};
ticks=[-1: 2/12:1];
f=figure; plot(t_sec,f_log2,'.');
title('Heidi Fundamental Frequency')
xlabel('Time (s)'); ylabel('Pitch');
for i=-1:1/6:1,
    line([0 15],[i i],'Color','k');
end
for i=-1:1:1,
    line([0 15],[i i],'Color','r');
end
set(gca,'YLim',[-1 1],'YTick',ticks,'YTickLabel',labs)
```


Heidi Fundamental Frequency
Pitch
a#/bb2
g#/ab2
f#/gb2
e2
d2
c2
a#/bb1
g#/ab1
f#/gb1
e1
d1
c1
a#/bb
0
5
10
15
Time (s)

Aliens in the Alps III

- Then they plot the histogram of the data with various bin positions to see some structure.

```
hist(f0oct,[-1:1/12:1])
xlabs={'a#/bb','c^1','d^1','e^1','f#/gb^1','g#/ab^1','a#/bb
    ^1','c^2','d^2','e^2','f#/gb^2','g#/ab^2','a#/bb^2',};
set(gca,'XLim',[-1 1],'XTick',[-1: 2/12:1],'XTickLabel',
    xlabs)
xlabel('Pitch'); ylabel('Frequency of Occurrence');
title('Heidi: Pitch Histogram'); xticklabel_rotate([],90,[])
```

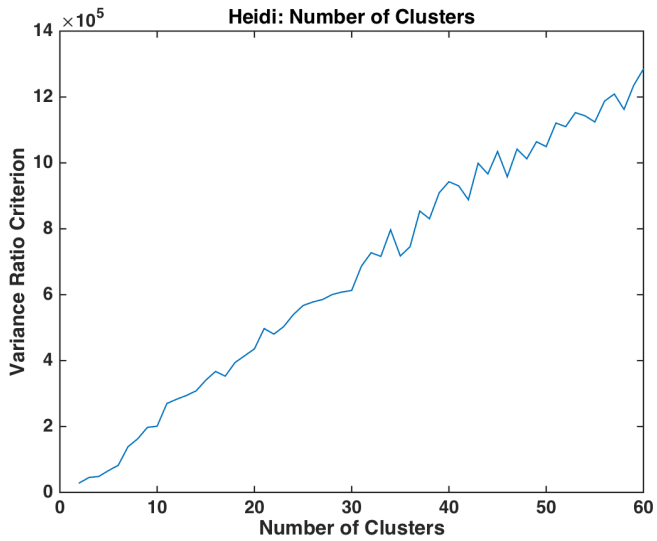


Aliens in the Alps IV

- Use the variance ratio criterion to determine how many different tones Heidi sings.

Aliens in the Alps V

```
eva = evalclusters(f0oct','
    kmeans','CalinskiHarabasz'
    ,'KList',[1:40])
figure;plot(eva.
    CriterionValues)
```

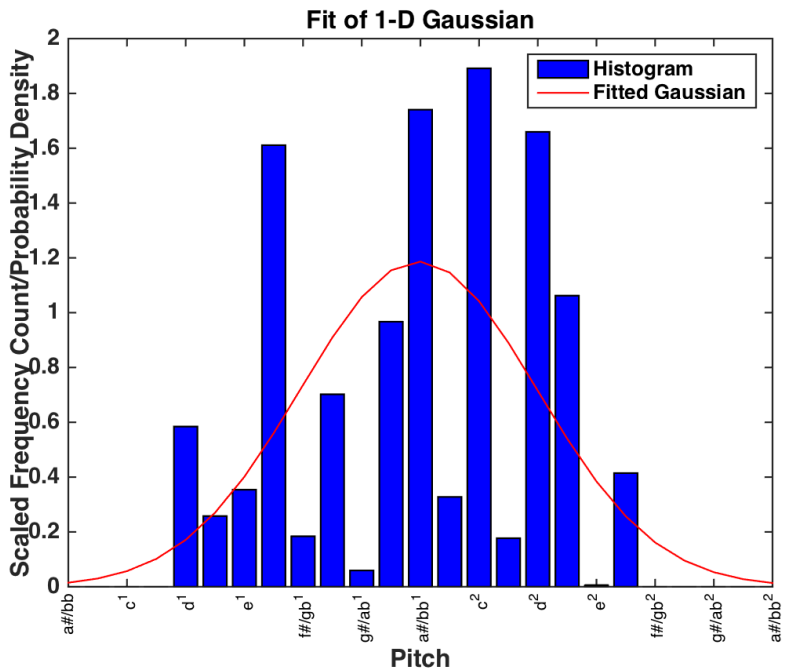


Does not look so good!

Aliens in the Alps VII

Let us try to fit a Gaussian to the frequency histogram. To make sure that we can compare the fitted normal probability density function and the histogram, devide the the area under the histogram (integral) by the number of frequency points times the difference between two adjacent bins: `(length(f_log2)*(b(2)-b(1)))`

```
  m=mean(f_log2); s2=var(f_log2)
b=[-1:1/12:1]; h=hist(f_log2,b);
n0=pdf('norm',b,m,sqrt(s2));
h0=h/(length(f_log2)*(b(2)-b(1)));
bar(b,h0,'b'); hold on; plot(b,n0,'r')
set(gca,'XLim',[-1 1],'XTick',[-1: 2/12:1],'XTickLabel',
    xlabs)
```



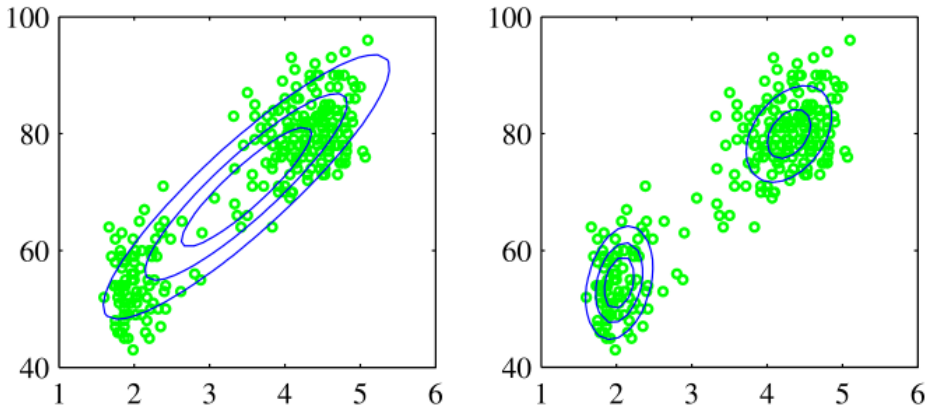
Gaussian Mixture Model (GMM)

Gaussian Mixture Model (GMM): Applications

- Speaker recognition (preprocessing before a Bayesian model, e.g. Hidden Markov model)
- Speech recognition (preprocessing before a Bayesian model, e.g. Hidden Markov model)
- Pricing of stock options
- Preprocessing for image processing (image classification)

Mixture of Gaussians

- It is not possible to represent complex, multimodal data distributions accurately using a single Gaussian.
- Example: The sound of one instrument $\rightarrow$ One Gaussian
  The sound of many instruments of the same kind $\rightarrow$ Mixture of Gaussians



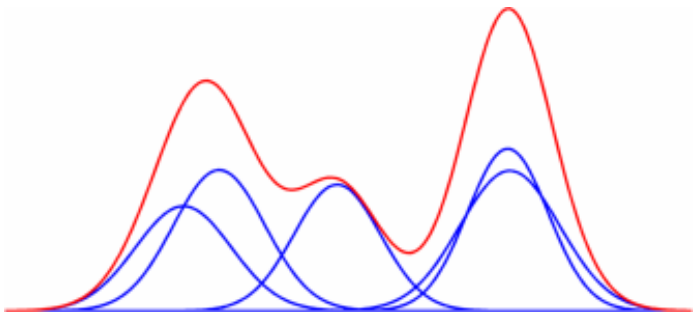
Gaussian Mixture Model

- A mixture of K Gaussians is given by

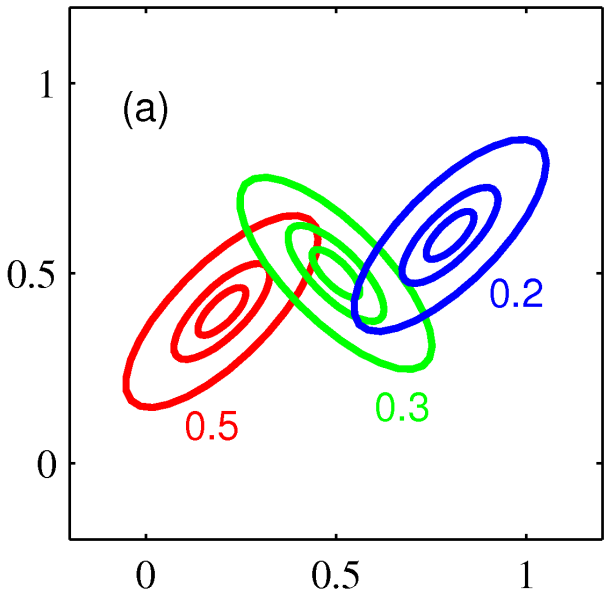
$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \mathbf{\Sigma}_k)$$

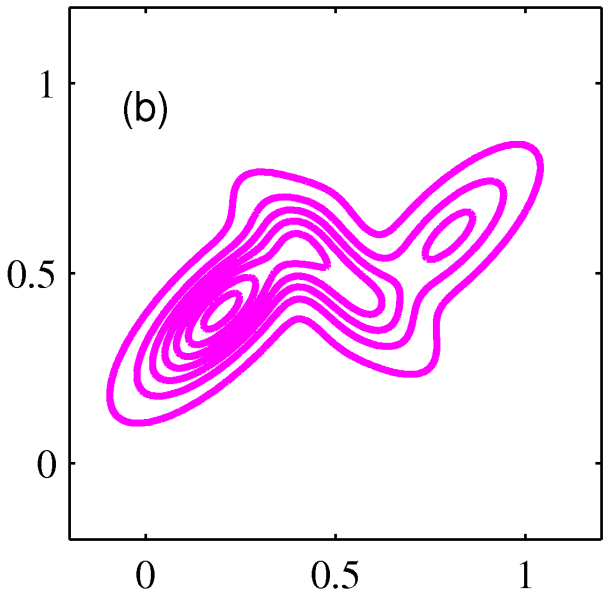
- The parameters π_k are called *mixing coefficients* and satisfy the properties

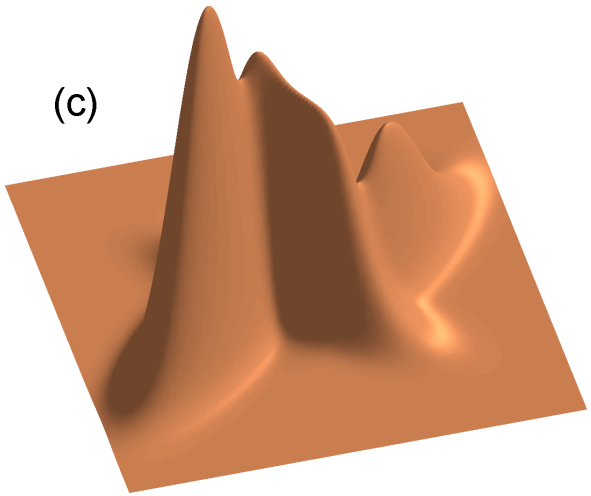
$$\sum_{k=1}^{K} \pi_k = 1, 0 \leq \pi_k \leq 1$$



Example of GMM with 3 Mixture 2 D Components







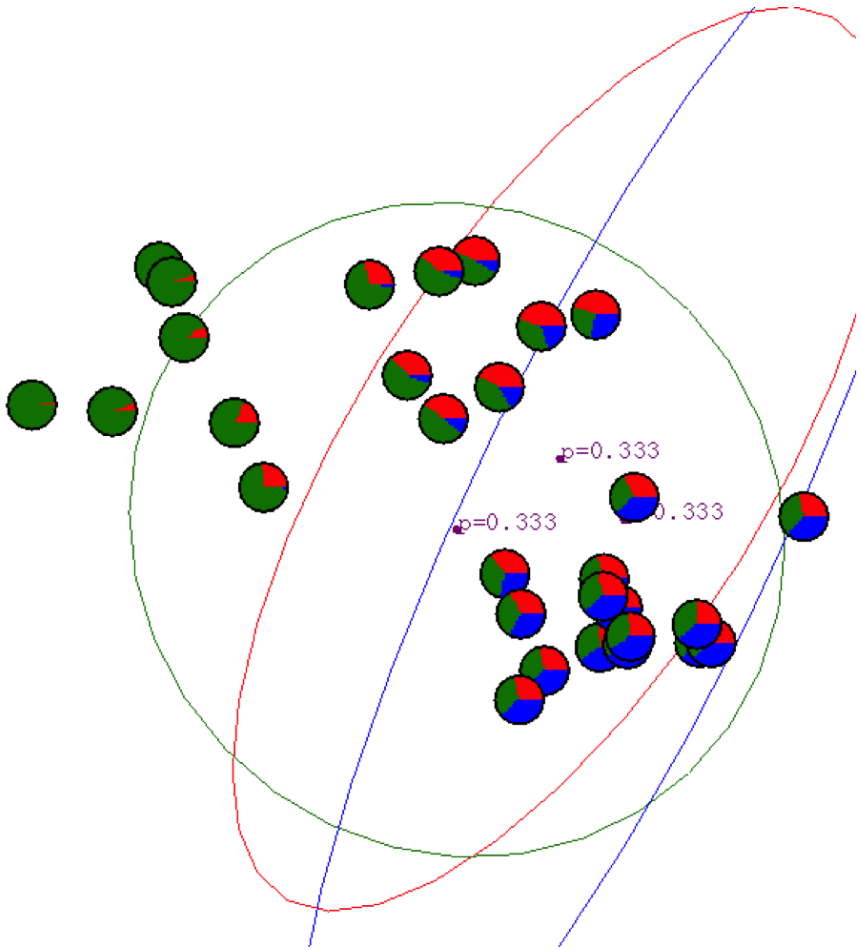
Applications of GMMs

- Method for clustering and density estimation
- Example applications:
  - Phoneme classification in automatic speech recognition (e.g. as a preprocessing stage for a hidden Markov model).
  - Vocal / non-vocal detection: model spectral distribution of various vocal tract configurations by a combination of Gaussian components.
  - Automatic singer identification
  - Instrument classification.

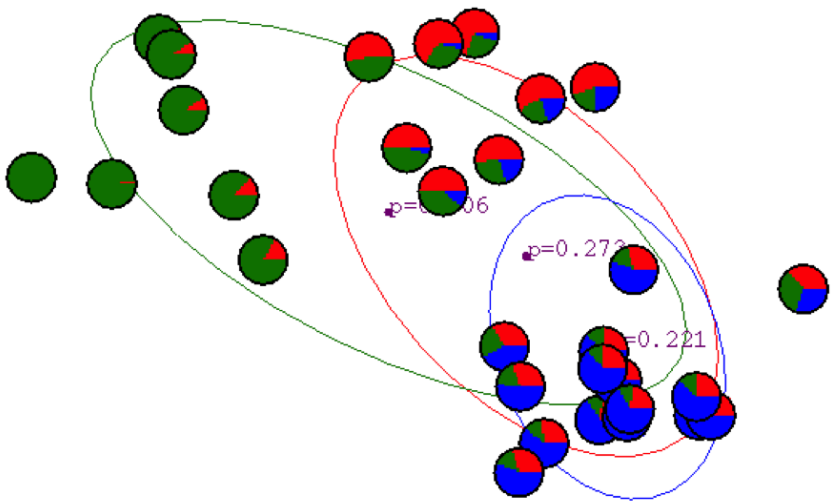
Soft Clustering

- In k-means or agglomerative clustering from last lecture points $\mathbf{x}$ belong to exactly one cluster k out of K different clusters.
- In *soft membership clustering*, a point $\mathbf{x}$ can belong to different clusters k, expressed by a probability $p(k|\mathbf{x})$.
- (For hard clustering $p(k|\mathbf{x}) = 1$ for one cluster k and $p(j|\mathbf{x}) = 0$ for all others $j \neq k$)
- How to calculate the soft membership?

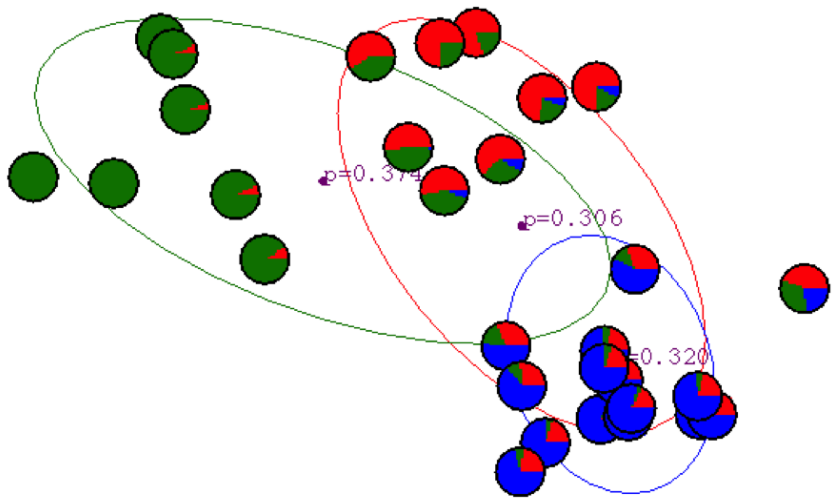
Example GMM EM Training: Start



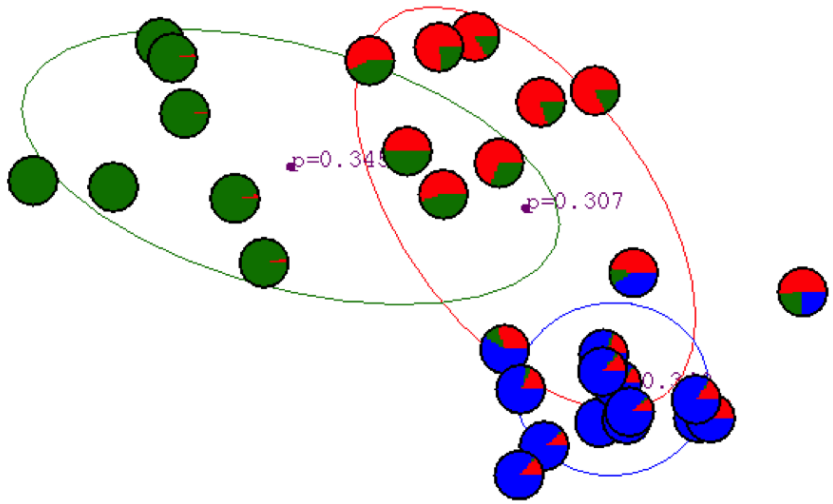
Example GMM EM Training: iteration 1



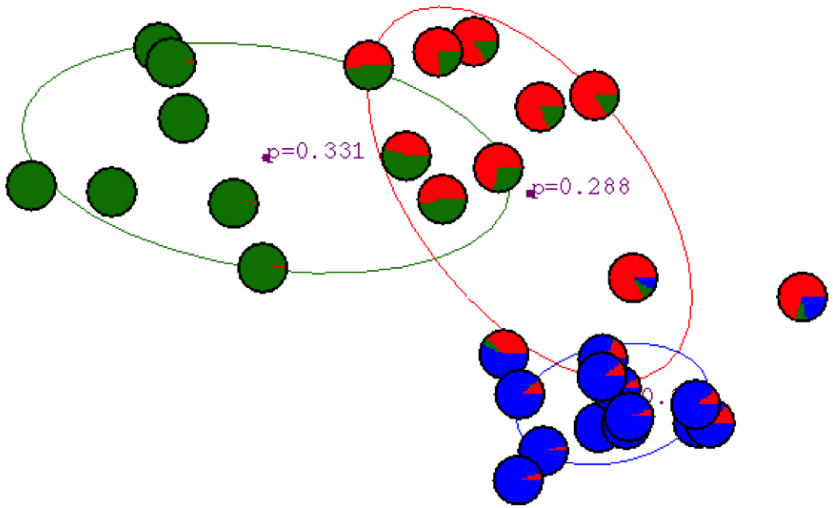
Example GMM EM Training: iteration 2



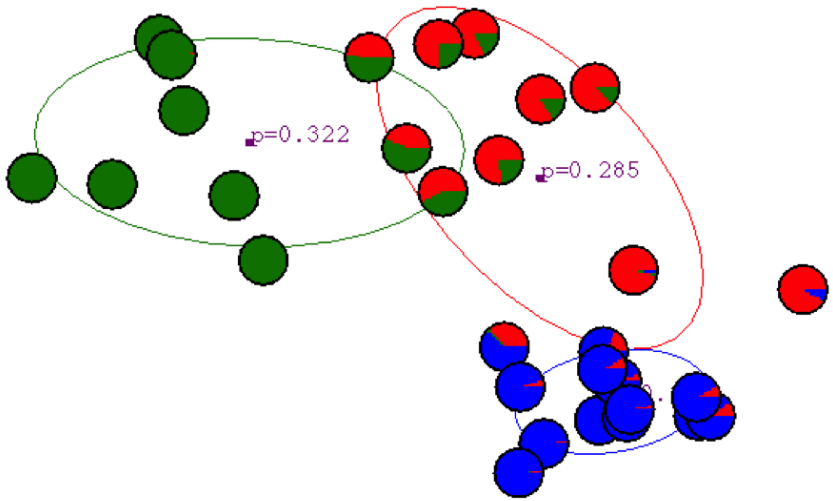
Example GMM EM Training: iteration 3



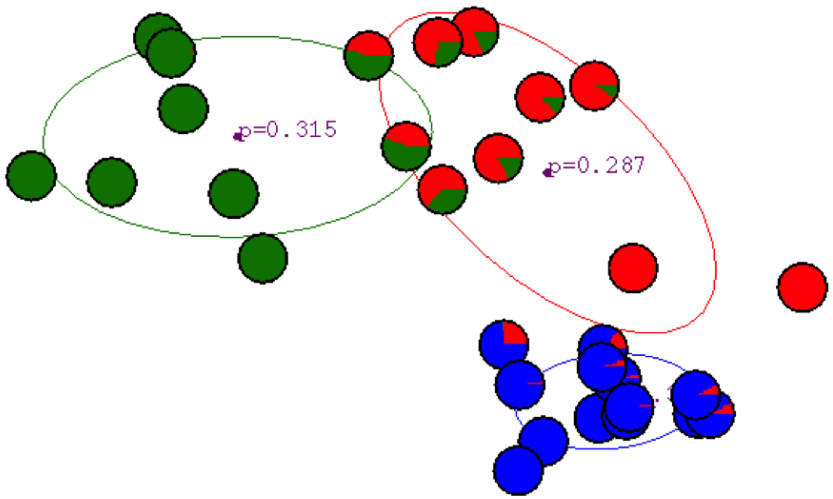
Example GMM EM Training: iteration 4



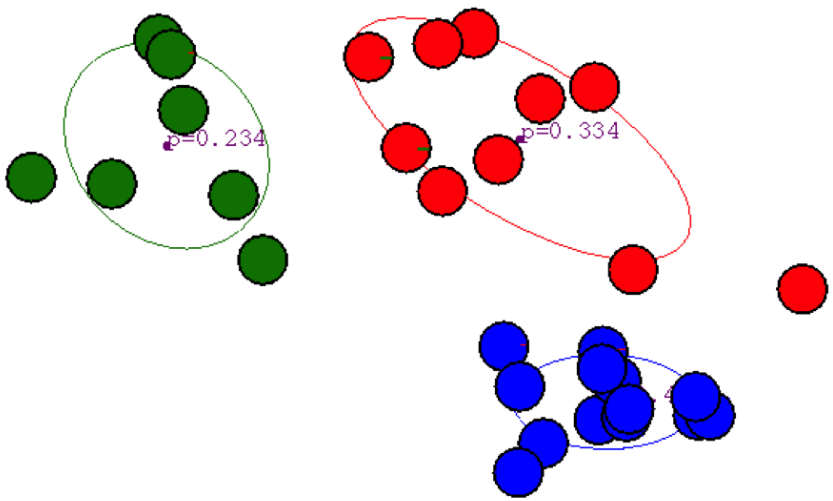
Example GMM EM Training: iteration 5



Example GMM EM Training: iteration 6



Example GMM EM Training: iteration 20



Gaussian Mixture Model: Parameters and Initialization

- Parameters:
  - $\bar{\mathbf{x}}_k$: estimate of mean of cluster k out of K clusters
  - $\mathbf{S}_k$: Estimate of covariance matrix for cluster k
  - Prior probability π_k for cluster k
  - Points $\mathbf{x}_i$ in cluster k normally distributed according the probability density function:
$$\mathcal{N}(\mathbf{x}|\bar{\mathbf{x}}_k, \mathbf{S}_k)$$
  - Parameters altogether: $((\pi_1, \bar{\mathbf{x}}_1, \mathbf{S}_1), (\pi_2, \bar{\mathbf{x}}_2, \mathbf{S}_2), \ldots, (\pi_K, \bar{\mathbf{x}}_K, \mathbf{S}_K))$

(1) Initialization:

- $\bar{\mathbf{x}}_k, \mathbf{S}_k, pi_k$ initialized by random or a smart guess, e.g. k-means for estimating means and in addition covariances for each cluster k.

GMM II: Expectation Step

(2) Calculate for all I points and all K clusters the conditional probability for a point $\mathbf{x}_i$ to belong to cluster k:

$$
\begin{aligned}
p(k|\mathbf{x}_i) &= \frac{p(k)p(\mathbf{x}_i|k)}{p(\mathbf{x}_i)} \\
&= \frac{p(k)p(\mathbf{x}_i|k)}{\sum_{j=1}^{K} p(j)p(\mathbf{x}_i|j)} \\
&= \frac{\pi_k \mathcal{N}(\mathbf{x}_i|\bar{\mathbf{x}}_k, S_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_i|\bar{\mathbf{x}}_j, S_j)}
\end{aligned}
$$

Relevant terms:

- Bayes (posterior: $p(k|\mathbf{x})$, likelihood: $p(\mathbf{x}|k)$, prior: $p(k)$, normalization: $p(\mathbf{x})$)

$$p(k|\mathbf{x}) = \frac{p(\mathbf{x}|k)p(k)}{p(\mathbf{x})}$$

- Marginalization:

$$p(\mathbf{x}_i) = \sum_{j=1}^{K} p(j)p(\mathbf{x}_i|j)$$

GMM III: Maximization Step (left)

(3) Sample count, mean, cov, prior in soft clustering

- Soft count: $N_k = \sum_{i=1}^{I} p(k|\mathbf{x}_i)$
- Soft sample mean: $\bar{\mathbf{x}}_i = \frac{1}{N_k} \sum_{i=1}^{I} p(k|\mathbf{x}_i)\mathbf{x}_i$
- Soft sample covariance matrix: $\mathbf{S}_k = \frac{1}{N_k} \sum_{i=1}^{I} p(k|\mathbf{x}_i)(\mathbf{x}_i - \bar{\mathbf{x}}_i)(\mathbf{x}_i - \bar{\mathbf{x}}_i)^T$
- Prior: $\pi_k = \frac{N_k}{I}$

Comparison: Sample count, mean, cov, prior for 'hard' k-means, LDA, QDA:

- Sample mean (e.g. k-means, LDA, QDA): $\bar{\mathbf{x}}_i = \frac{1}{I_k} \sum_{i=1}^{I} p(k|\mathbf{x}_i)\mathbf{x}_i$
- Sample class covariance (e.g. in QDA): $S_k = \frac{1}{I_k - 1} \sum_{i=1}^{I} (\mathbf{x}_i - \bar{\mathbf{x}}_i)(\mathbf{x}_i - \bar{\mathbf{x}}_i)^T = \frac{1}{I_k - 1} \mathbf{X}_c^T \mathbf{X}_c$
- Class prior: $\pi_k = \frac{I_k}{I}$

GMM EM Evaluation

(4) Evaluation of likelihood

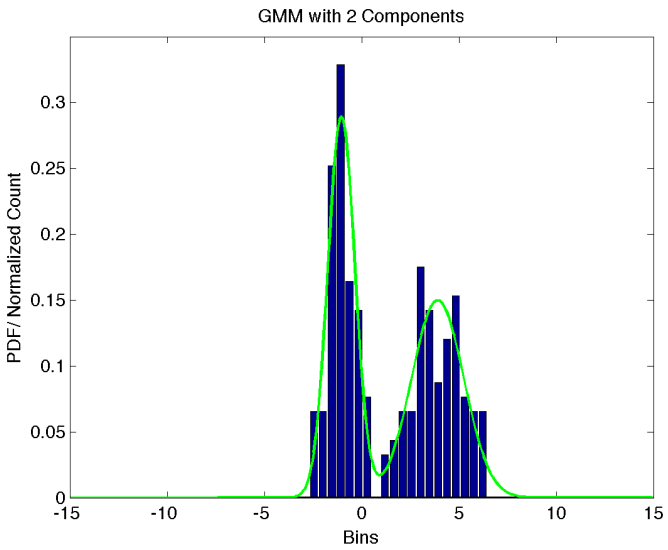
$$p(\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_I | ((\pi_1, \bar{\mathbf{x}}_1, \mathbf{S}_1), (\pi_2, \bar{\mathbf{x}}_2, \mathbf{S}_2), \ldots, (\pi_K, \bar{\mathbf{x}}_K, \mathbf{S}_K))) = \Sigma_{i=1}^{I} \Sigma_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_i | \bar{\mathbf{x}}_k, \mathbf{S}_k)$$

- Check for convergence of p or parameters. If convergence stop, otherwise return to step (2).

Further reading:

- Bishop pp. 430-439.
- Andrew Moore's machine learning tutorial lectures: Gaussian Mixture Models, `http://www.autonlab.org/tutorials/`

GMM in Matlab



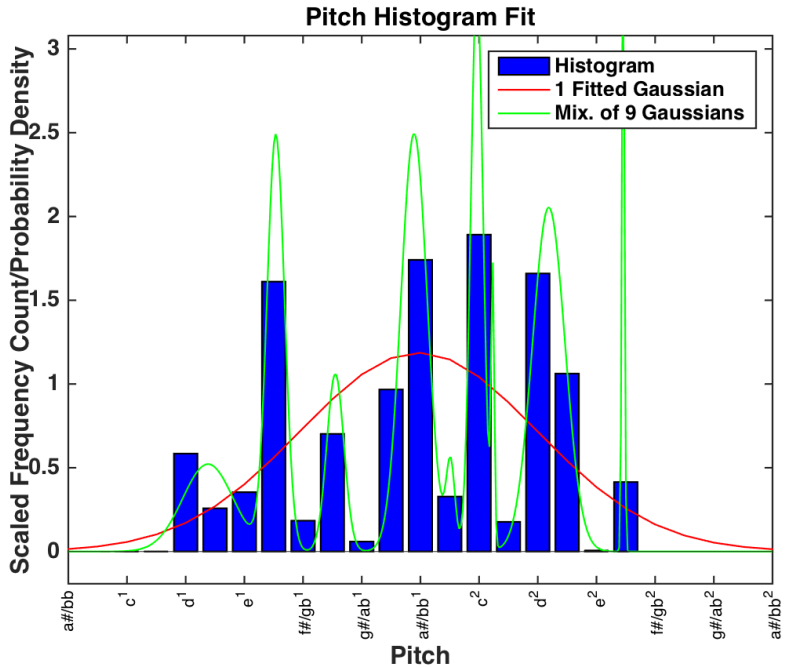

```
m1=-1, s1=0.5, m2=4, s2=2;
x=[mvnrnd(m1,s1,100);
mvnrnd(m2,s2,100)]; [h b] =hist(x
    ,20);
h_norm=h/(length(x)*(b(2)-b(1)));
bar(b,h_norm); hold on; no_gauss=2;
opt=statset('MaxIter',100);
w=fitgmdist(x,no_gauss,'Options',opt
    ,'Replicates',10)
h=ezplot(@(x)pdf(w,x),[-15 15])
set(h,'Color','g');
set(gca,'YLim',[0 0.4],'XLim',[-15
    15])
```

Aliens in the Alps

Class Assignment

Fit a Gaussian Mixture Model with 9 component to Heidi's frequency data. If you get a warning saying `Warning: Failed to converge in 3 iterations for gmdistribution with 9 components` *try again possibly with more iterations.*

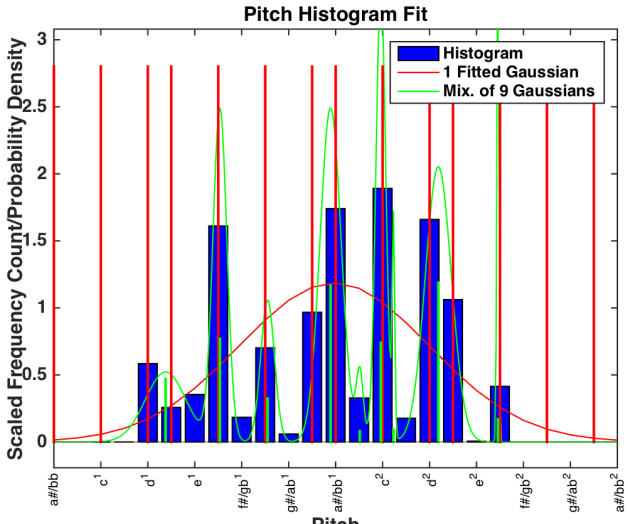
```
opt=statset('MaxIter',300);
w=fitgmdist(f_log2',9,'Options',opt,'Replicates',20)
h=ezplot(@(x)pdf(w,x),[-1 1])
```



Class Assignment

Now the aliens find a book about Earthling's music theory, where one can find that a major scale has notes on the following fractions of an octave: $0, \frac{2}{12}, \frac{4}{12}, \frac{5}{12}, \frac{7}{12}, \frac{9}{12}, \frac{11}{12}$ *Plot these pitches as vertical lines into the previous plot. Compare these pitches with the positions of the means of the 9 Gaussians of the GMM.*

```
majscale=[0 2 4 5 7 9 11]; ticks=[(majscale-12)/12 majscale
    /12]; labs={'bb','c^1','d^1','eb^1','f^1','g^1','a^1','
    bb^1','c^2','d^2','eb^2','f^2','g^2','a^2'};
for i=1:length(ticks),
    line([ticks(i) ticks(i)],[0 2.8],'Color','r');
end
for i=1:9,
    line([w.mu(i) w.mu(i)],[0  w.ComponentProportion(i)*5],'
    Color','g');
end
```



Summary GMM

- For music/speech/speaker recognition GMMs are often combined with a time series analysis model (e.g. Hidden Markov Model) and form an example of Bayesian models.

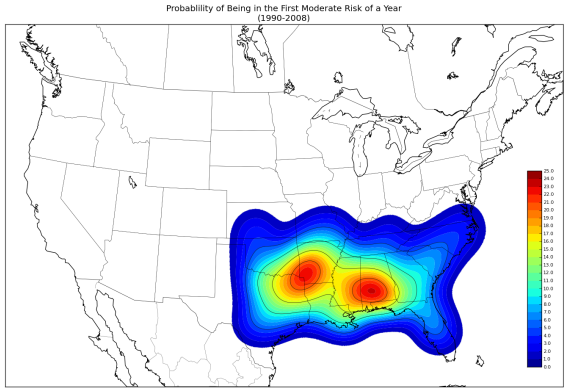
Kernel Smoothing

Kernel Smoothing

- A statistical technique for estimating a real valued function f(X) $X \in \mathbb{R}^p$, in particular a probability density function, by using its noisy observations, when no parametric model (e.g. fitting a Gaussian by estimating its parameters mean μ and variance σ^2) for this function is known.
- The estimated function is smooth, and the level of smoothness is set by the bandwidth parameter r
- After kernel smoothing, the set of irregular data points are represented as a smooth line or surface.
  `http://en.wikipedia.org/wiki/Kernel_smoother`

Applications of Kernel Smoothing

- Of few customers we know the income. Do we have more customers with income around 300 000 DKK or 1 000 000 DKK?
- Estimation of crime rates in different areas before buying a house there
- Updating information in computer vision (on-line object tracking) as new data becomes available
- Stormrisks in the US (Image)



http://www.pmarshwx.com/blog/2011/02/03/aotw-storm-prediction-center-moderate-high-risks/

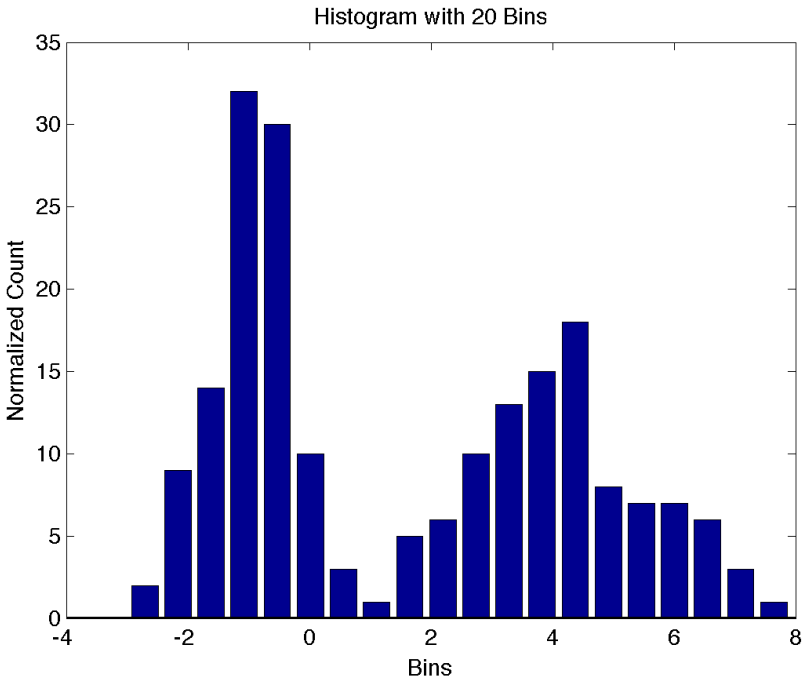
http://www.cs.utah.edu/~lifeifei/papers/kernelsigmod13.pdf

- A histogram partitions vector $\mathbf{x} = (x_1, x_2, \ldots, x_I)$ of I one-dimensional measurements into K distinct bins of band width r and then counts the number of observations falling in each bin.
- Define the *histogram* $\mathbf{h} = (h_1, \ldots, h_K)$ by

$$h_k = |\{x_i : |x_i - b_k| \leq \frac{r}{2}\}| (1 \leq k \leq K)$$

for K equally spaced *center bins* $b = (b_1, \ldots, b_K)$ and band width r.

Histogram: Matlab Example




```
 m1=-1, s1=0.5, m2=4, s2=2;
x = [mvnrnd(m1,s1,100);
mvnrnd(m2,s2,100)];
plot x in a histogram with 20
bins (2p) bin_no=20;
[ h b ] =hist(x,bin_no);
figure; bar(b,h);
xlabel('Bins')
ylabel('Normalized Count');
```

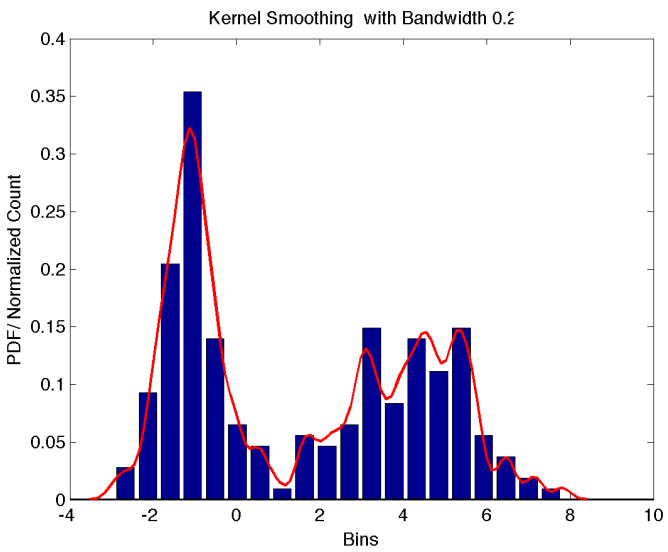
- In the Histogram, the choice of band width r is critical
  - a too high r can eliminate relevant peaks
  - a too small r can create spurious peaks in the pitch histogram.
- In kernel smoothing, using the Gaussian kernel function the smoothed histogram is calculated as:

$$h_k = \sum_{i=1}^{I} \frac{1}{\sqrt{2\pi}r} e^{-\frac{(x_i - b_k)^2}{2r^2}}.$$

for $k = 1, \ldots, K$.

- In the smoothed histogram, the Gaussian kernel replaces each single measurement x_i by a smooth Gaussian and then adds up all Gaussians across all points.
- *Bandwidth* r determines the smoothness of the histogram.
- Trade-off between noise sensitivity at small r and over-smoothing at large r values.
- Selection of the appropriate smoothing parameter r has to be guided by the task the histogram is used for.

Kernel Smoothing: Matlab Example



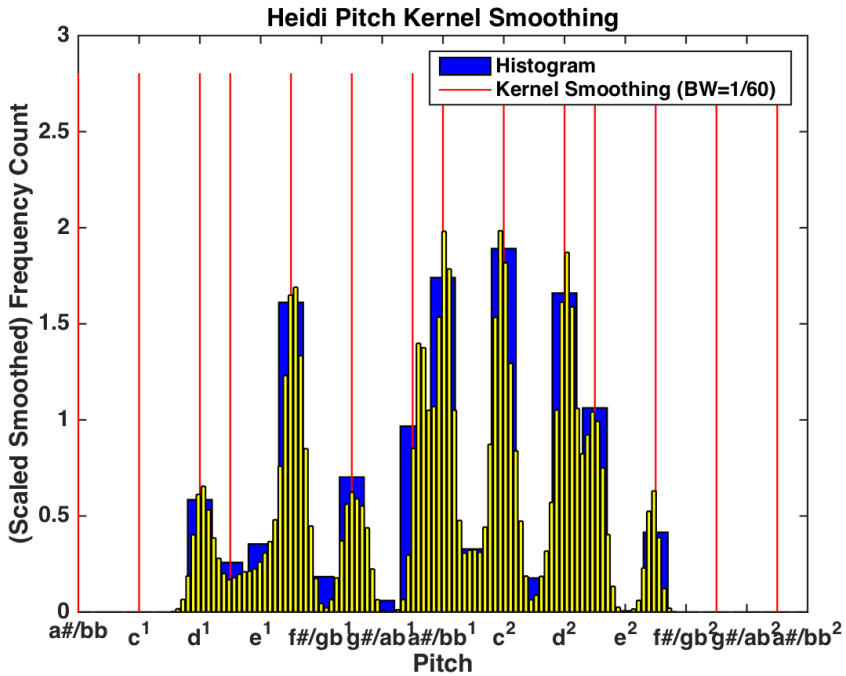

```
m1=-1, s1=0.5, m2=4, s2=2;
x = [mvnrnd(m1,s1,100);
mvnrnd(m2,s2,100)];
bw=0.2;
[h_ks,b_ks]=ksdensity(x,'bandwidth',
    bw);
[ h b]=hist(x,20);
h_norm=h/(length(x)*(b(2)-b(1)));
    figure; bar(b,h_norm); hold on;
plot(b_ks,h_ks,'r');
```

Aliens in the Alps

Class Assignment

Use kernel smoothing to capture the scale notes in Heidi's singing. Experiment with various bandwidths, in order so that all peaks in the smoothed histogram correspond to the scale notes apparently sung. Which scale note seemed to be sung differently than its theoretical pitch?

```
bw=1/60; [h_ks,b_ks,bw] = ksdensity(f_log2,'bandwidth',bw);
% plot ks result in yellow, bar heights scaled by 0.8 so we
    can still
% see the orig. histogram behind
bar(b_ks,h_ks*0.8,'y')
```



Exam

Exam I

- Individual oral exam
- Duration of exam: 15 minutes total for presentation, questions and grading.
- Mini project: Each student is required to complete an individual project in which the methods covered in the course are applied to a real-life classification problem.
- In the project, the student must apply multivariate statistics and pattern recognition to a problem.
- He/she must demonstrate that he/she understands the basic concepts and methods of multivariate statistics and pattern recognition.

Exam II

- Each student has a data set and must then solve the associated classification problem.
- In applying the methods from the course to the selected data, MATLAB must be used.
- The mini project must at least satisfy the following criteria:
  - Different classifiers must be tested and both parametric and non-parametric methods must be used.
  - All classifiers must be tested using cross-validation.
  - Feature reduction/selection must be considered.

Exam III

- Student will present the mini project to the examiner (Hendrik) and the censor (5 min.). It is important that the students practise the presentation and keep this time limits, since the exams are under a strict time schedule.
- Examiner and censor will ask questions about project and the course.
- Student must prepare a set of slides for presenting the project.
- The MATLAB code used in the project must be available at the exam and failure to comply will result in a failed grade.

Exam IV

- Exam date: January 6./7. (MED7) and 13. (SMC) 8h-18h
- Submit (a preliminary) set of slides and MATLAB code the day before the exam 12h noon over Moodle.
- Use pdf for the slides and zip for compressing (don't use `rar`)
- Final version of slides used in the presentation can vary slightly from the slides submitted on the day before.
- The student has to bring a laptop with the prepared slides, MATLAB installed and the MATLAB code for the project, ready to be connected to a projector. Make sure there are no problems with connection to the projector.

Mini Project

Data Sets

From the webpage `http://archive.ics.uci.edu/ml/` you will have to analyze one of the following data sets:

1. Letter Recognition
2. Breast Cancer Wisconsin (Original) Data Set
3. Wine
4. Ecoli
5. Seeds
6. Skin Segmentation
7. Parkinsons

Typical Analysis Steps in the Mini Project I

- Formulate your problem: What do you want to classify?
- Read data and transform it into the prtools data format. Deal with missing data (e.g. remove that object or feature). If a feature is string valued, transform it into numbers (e.g. 0/1).
- Explore the data, e.g. by plotting feature vectors pairwise against each other and calculating the correlation to see relations among the features.
- If there is only two possible labels convert the class labels also into numbers (e.g. 0/1) and correlate the features with the class labels (feature selection).
- Perform principal component analysis, visualize the scores in 2 dimensions (the eigenvectors with highest eigenvalues). Discuss how the eigenvectors are composed of the original features. How many eigenvectors do you choose and why? Give the percentage of preserved variance when using this number of eigenvectors.

Typical Analysis Steps in the Mini Project II

- Possibly [apply clustering and calculate the optimal number of clusters.]
- Apply
  - parametric classifiers (e.g. quadratic discriminant analysis `qdc`, linear discriminant analysis `ldc`, minimum distance classifier `nmsc`)
  - non-parametric classifiers (e.g. k-nearest neighbor classifier `knnc`, [a neural net, a support vector machine]).
  - Possibly apply feature selection.
  - Possibly perform classification on the scores on the most prominent eigenvectors or on the selected features (that might be especially relevant if you have a high number of features).
  - Use cross-validation for the classification.

Typical Analysis Steps in the Mini Project III

- Display the confusion matrix for each classifier.
- Compare the classifiers with an appropriate evaluation measure (e.g. accuracy).
- If your problem is a 2 class problem you can also calculate the f-measure and the AUC, including a ROC plot.
- If possible compare with a baseline performance value (e.g. random classification or a classifier that always predicts the same class, no matter what the true class is).
- Discuss the results (possibly [speculate why which classifier performed better and how that might relate to the number of data points, features, the separability of the classes, the complexity/number of estimated parameters of the algorithm])