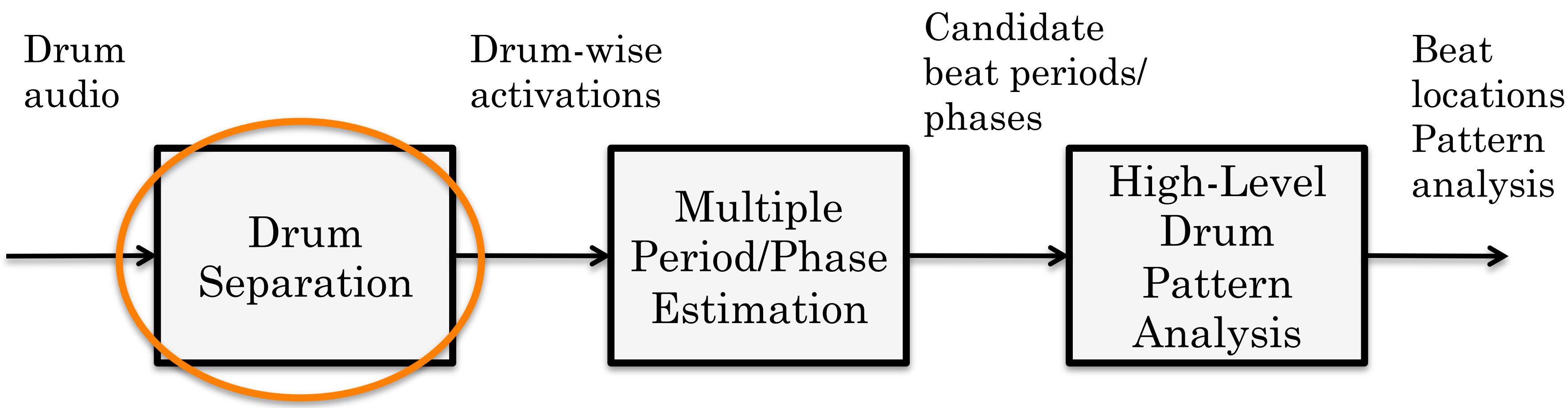
Toward Live Drum Separation Using Probabilistic Spectral Clustering Based on the Itakura-Saito Divergence

Eric Battenberg
UC Berkeley, Dept. of EECS
Parallel Computing Laboratory
CNMAT

TOWARD COMPREHENSIVE RHYTHMIC UNDERSTANDING

- Or "Live Drum Understanding"
- Goal: Go beyond simple beat tracking and provide context-aware, instrument-aware information in real-time, e.g.
  - "This rhythm is in 5/4 time"
  - "This drummer is playing syncopated notes on the hi-hat"
  - "The ride cymbal pattern has a swing feel"
  - "This is a Samba rhythm"

LIVE DRUM UNDERSTANDING SYSTEM



- <u>This work.</u>
- Gamma Mixture Model training of drum templates
- Non-negative decomposition onto templates.

✓

- HMM-based multi-hypothesis beat tracking.

✓

- Statistical deep learning of drum patterns
- Stacked Conditional Restricted Boltzmann Machines

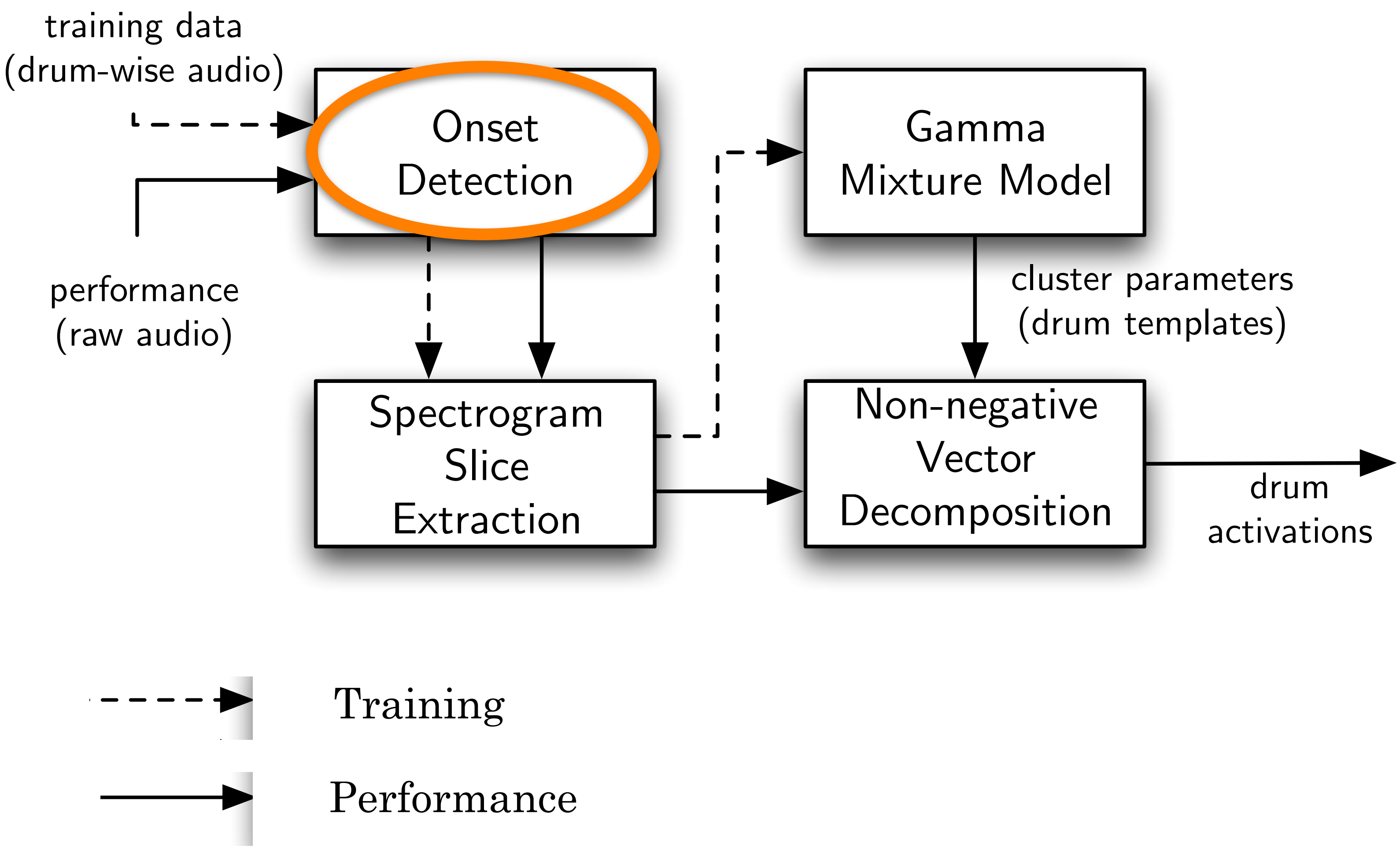
Requirements for Drum Separation

- Real-Time/Live operation
- Useful with <u>any</u> percussion setup.
  - Before a performance, we can quickly train the system for a particular percussion setup.

The Primary Takeaway

- Gamma Mixture Model
  - *For learning spectral drum templates.*
  - **Cheaper to train** than GMM
  - **More stable** than GMM
- Non-negative Vector Decomposition (NVD)
  - *For computing template activations from drum onsets.*
  - Learning **multiple templates per drum** improves separation.
  - The use of **“tail” templates** reduces false positives.

Drum Separation System

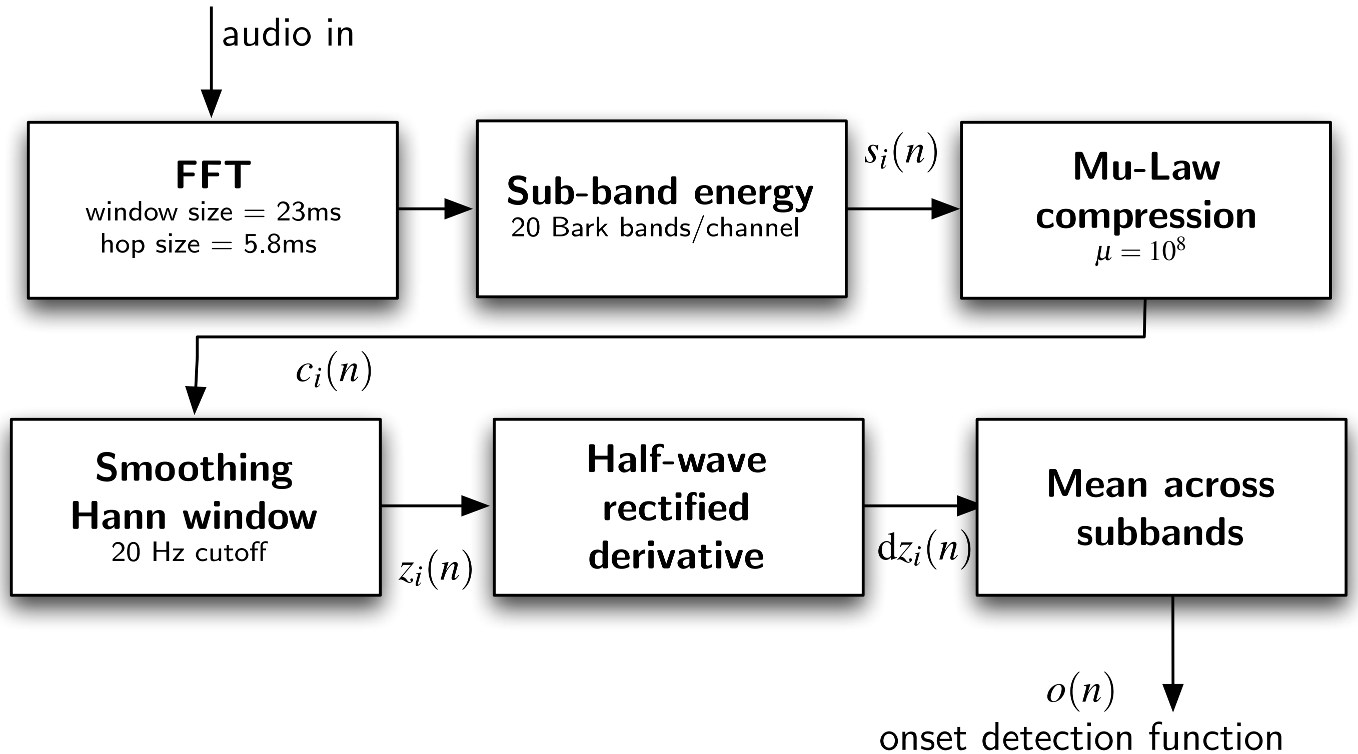


Onset Detection

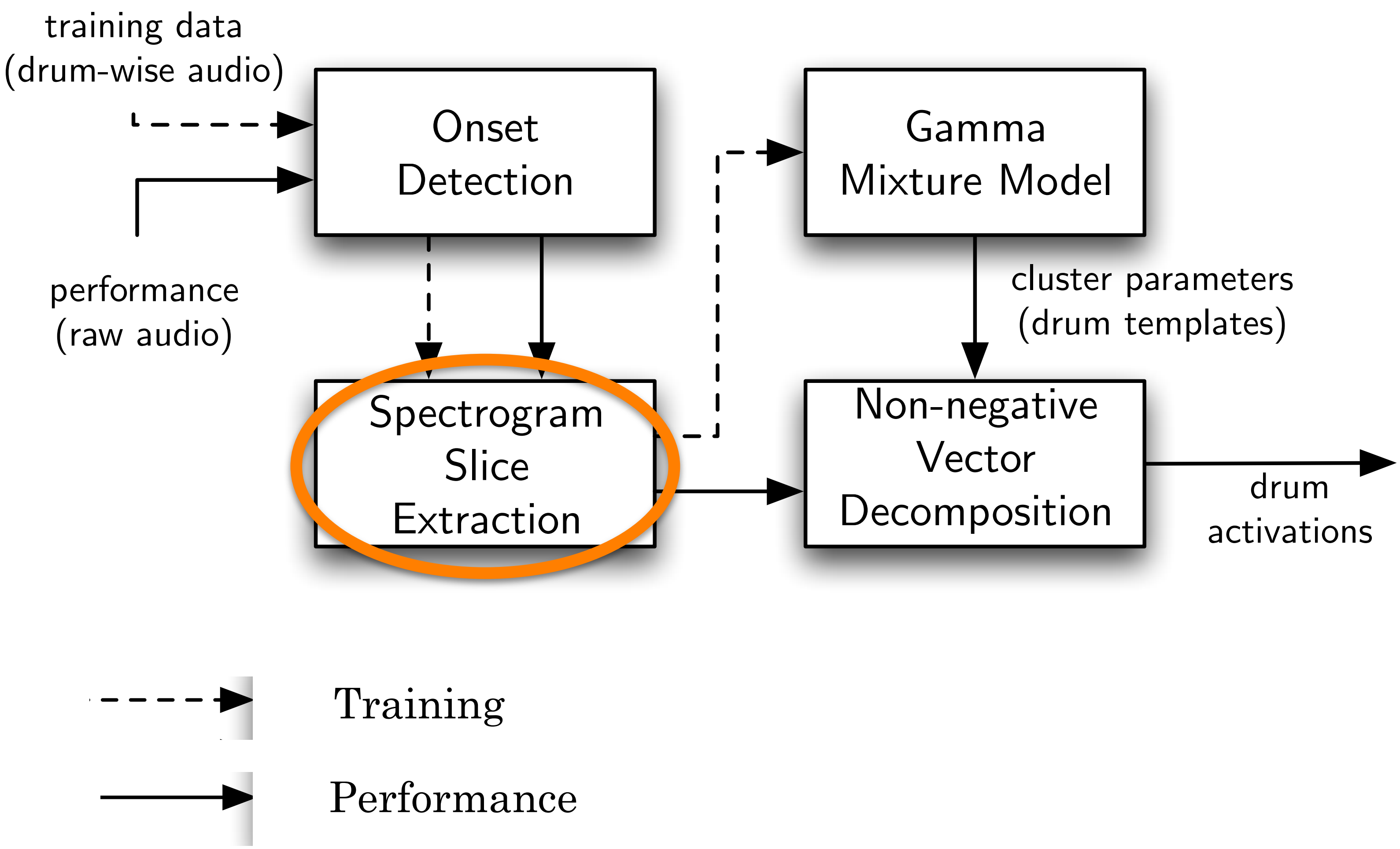
•Detection function: Differentiated log-energy of multiple perceptual sub-bands.

•On 2400 drum strikes, our **adaptive threshold** achieves:

- **85% recall, 99.9% precision**.

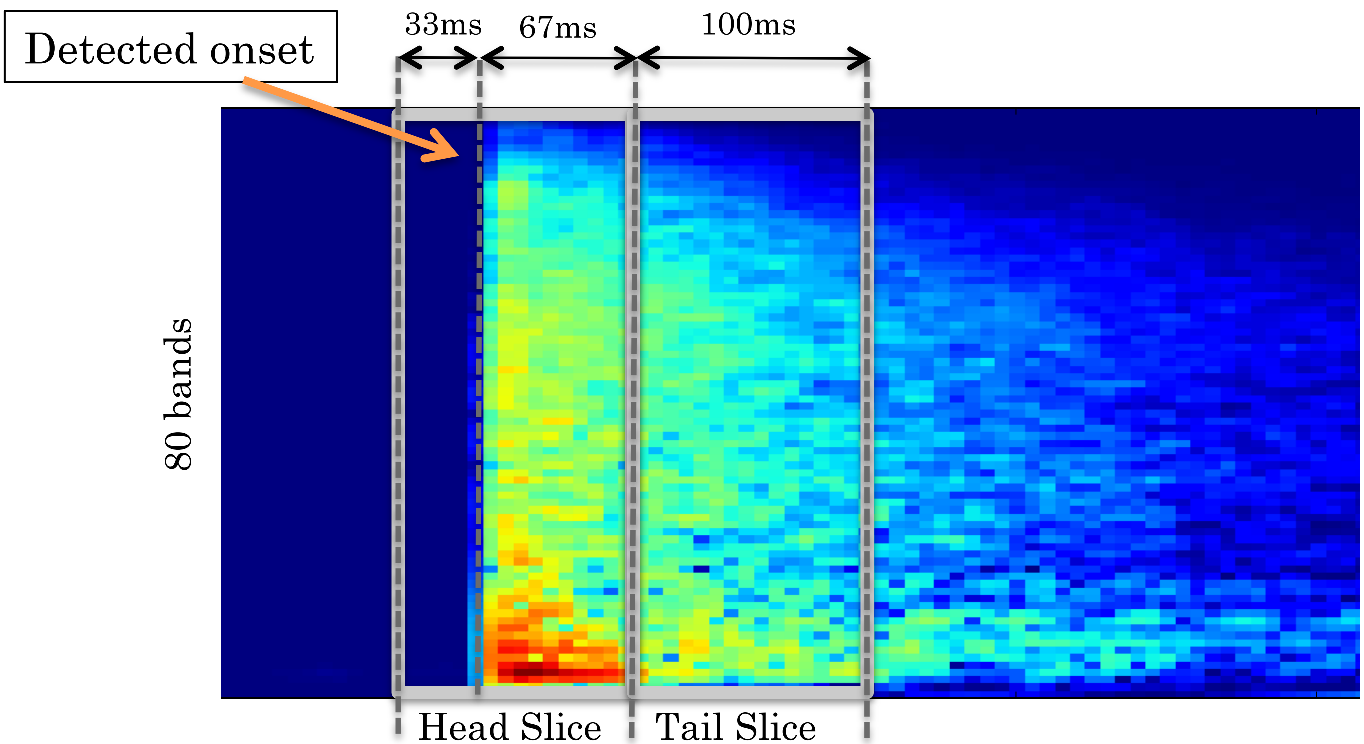


Drum Separation System

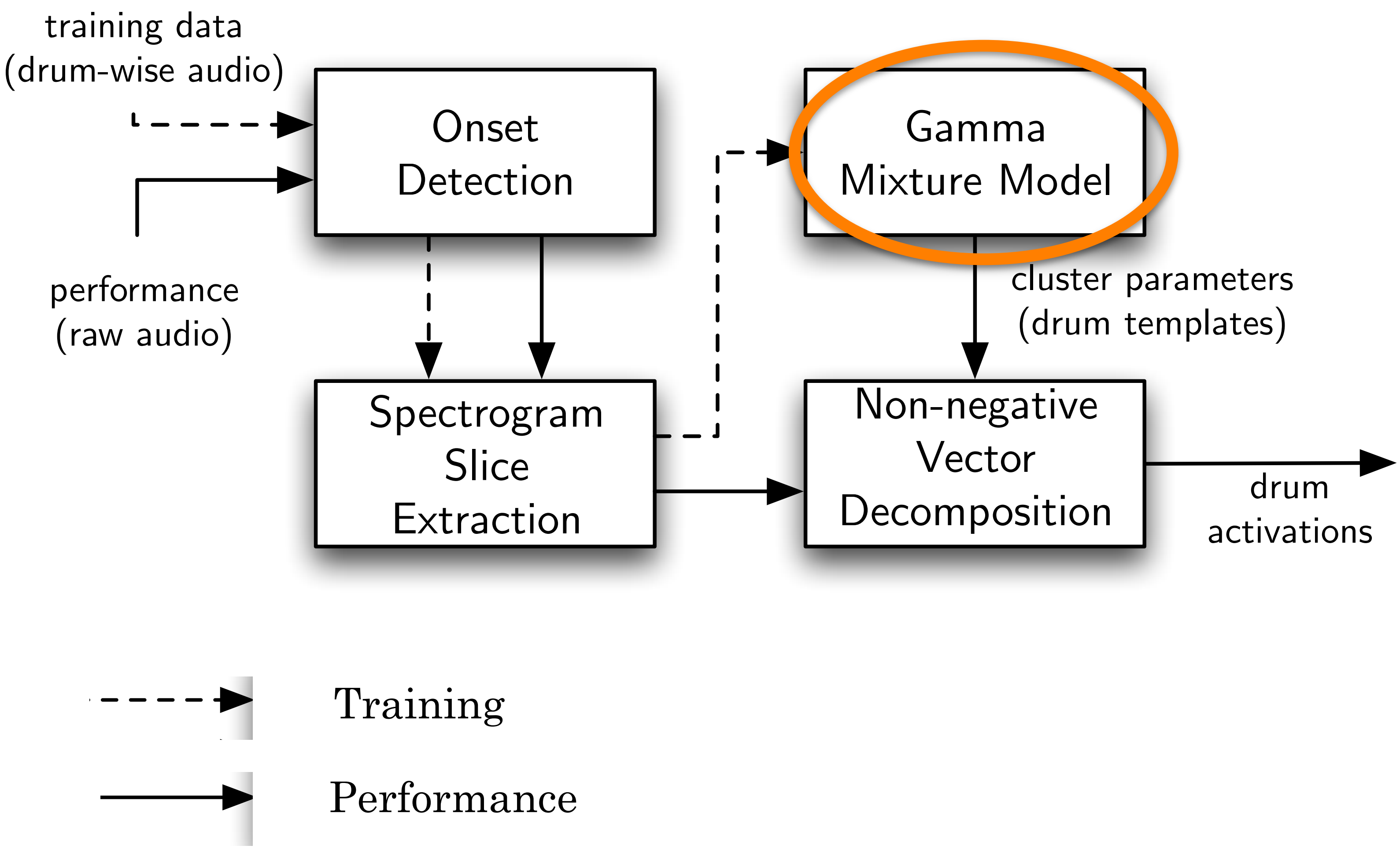


Spectrogram Slices

- Extracted at onsets.
- Each slice contains 100ms (~17 frames) of audio
- 80 bark-spaced bands per channel [Battenberg 2008]
- During training, both "head" and "tail" slices are extracted.
  - Tail templates serve as decoys during non-negative vector decomposition.

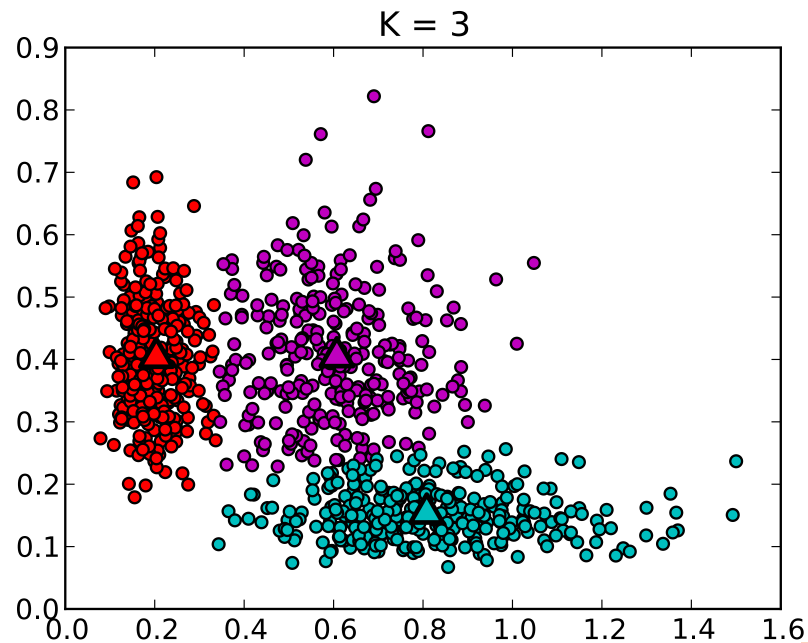


Drum Separation System



Training Drum Templates

- Instead of taking an "average" of all training slices for a single drum…
- Cluster them and use the cluster centers as the drum templates.
  - This gives us multiple templates per drum…
  - Which helps represent the variety of sounds that can be made by a single drum.



Clustering using Mixture Models

- Train using the Expectation-Maximization (EM) algorithm.
- Gaussian Mixture Model (GMM)
  - Requires expensive/unstable covariance matrices
  - Enforces a Euclidean distance measure.

$$d_{Euc}(X,Y) = \int_{\omega} (X(\omega) - Y(\omega))^2 \, d\omega$$

- Gamma Mixture Model
  - Single mean vector per component
  - Enforces an Itakura-Saito (IS) distance measure
    - A scale-invariant perceptual distance between audio spectra.

$$d_{IS}(X,Y) = \int_{\omega} \left[\frac{X(\omega)}{Y(\omega)} - \log \frac{X(\omega)}{Y(\omega)} - 1 \right] d\omega$$

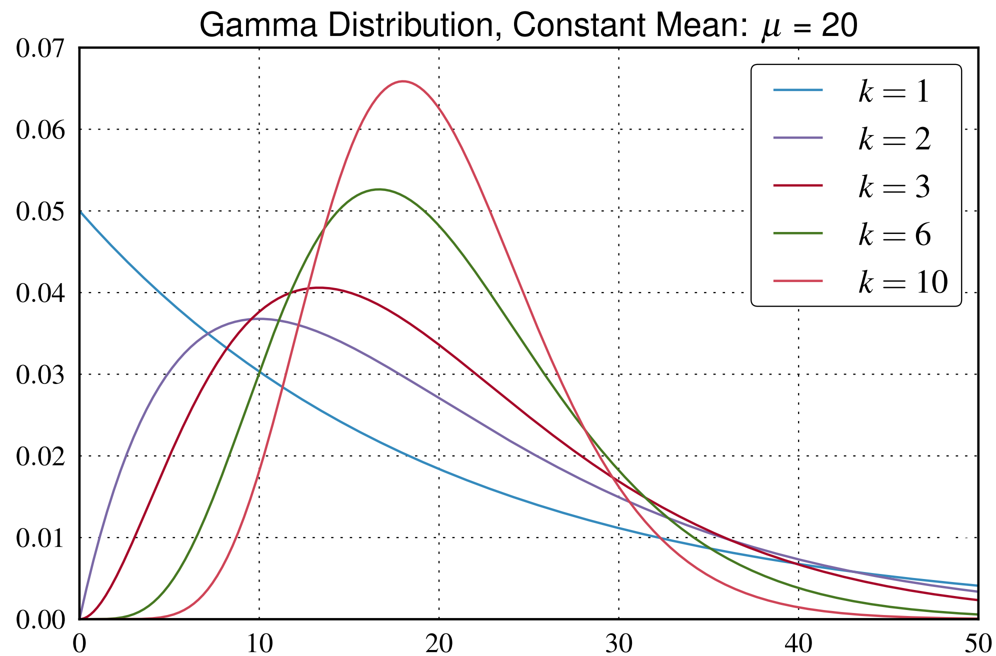
Gamma Distribution

- Our mixture model is composed of gamma distributions.
- The gamma distribution models the sum of k independent exponential distributions.

$$\mathrm{p}(y|\lambda,k) \quad = \quad y^{k-1}\frac{\lambda^k e^{-\lambda y}}{\Gamma(k)},$$

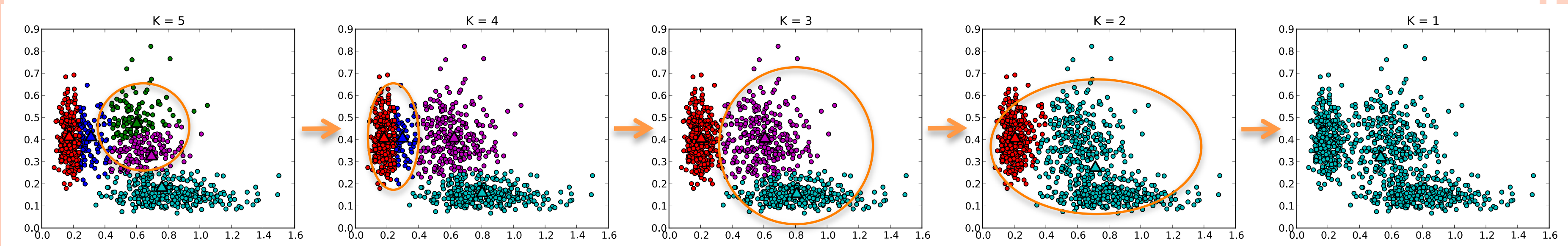
$$E[y] \quad = \quad \mu \quad = \quad k/\lambda$$
$$\mathrm{Var}[y] \quad = \quad \mu^2/k \quad = \quad k/\lambda^2$$

$$y \geq 0;\ \lambda, k > 0$$



Agglomerative Clustering

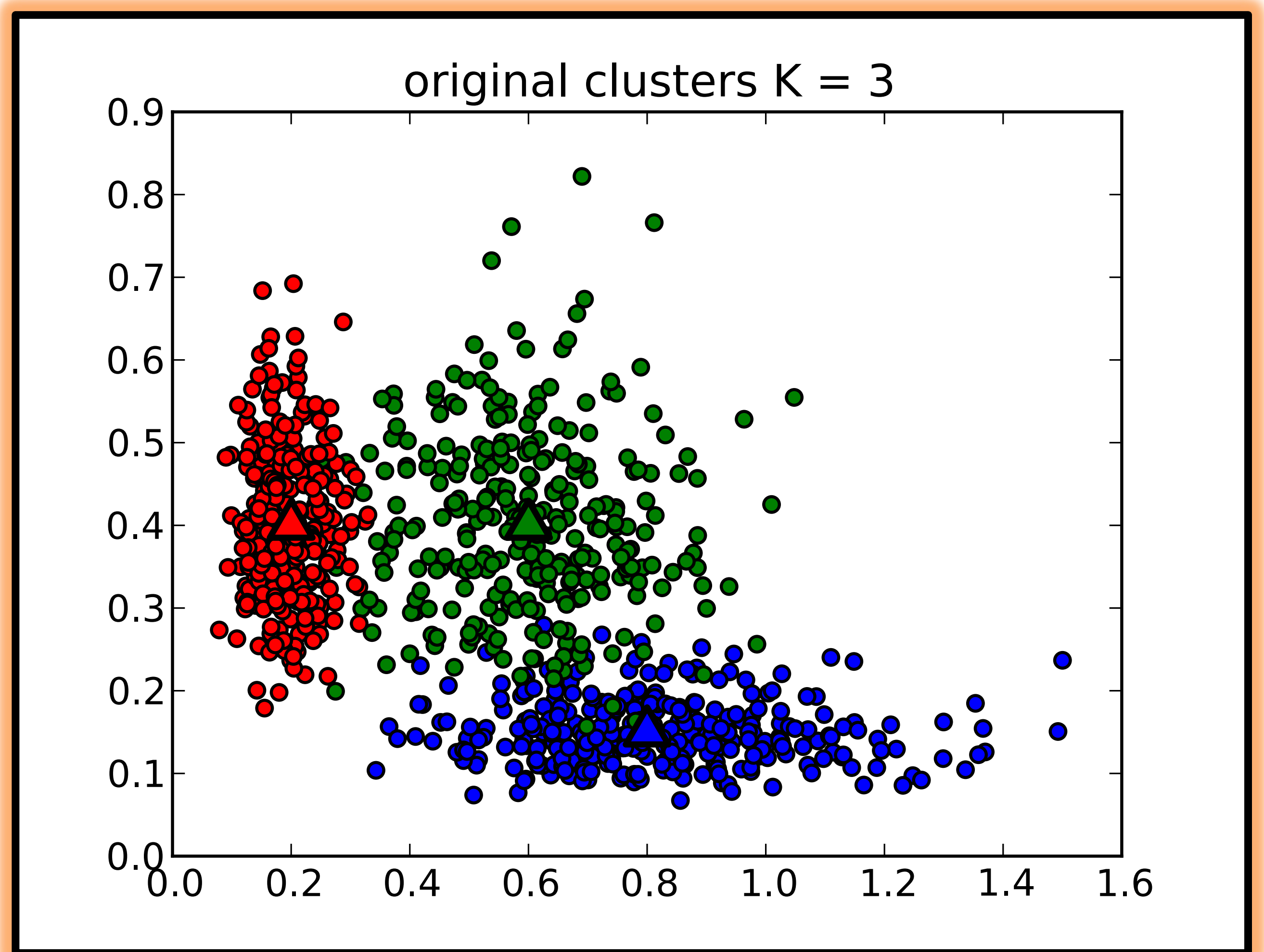
- *How many clusters to train?*
- We use Minimum Description Length (MDL) to choose the number of clusters.
  - Negative log-likelihood
  - + penalty term for number of clusters.



MDL= 187.8 MDL= -739.8 MDL= **-835.6** MDL= -201.5 MDL= 1373

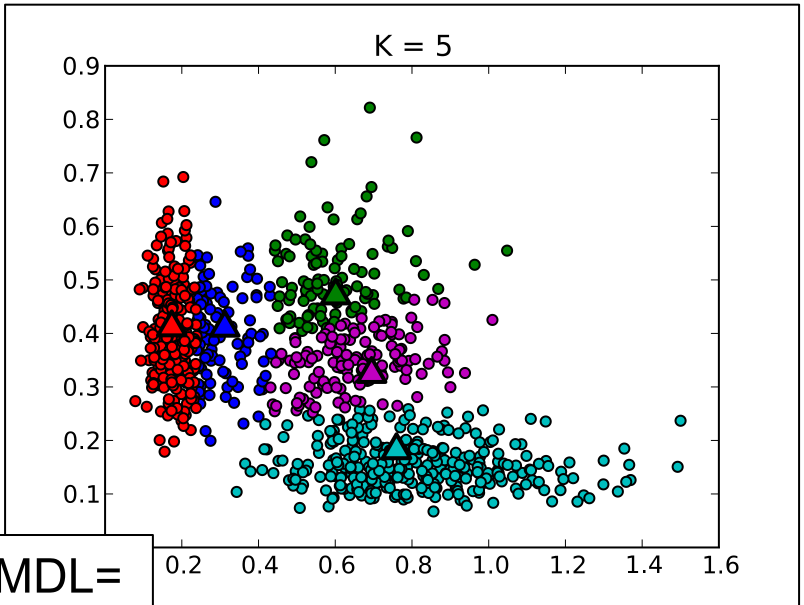
- 1. Run EM to convergence.
- 2. Merge the two most similar clusters.
- 3. Repeat 1,2 until we have a single cluster.
- 4. *Choose parameter set with smallest MDL.*

Agglomerative Clustering with MDL



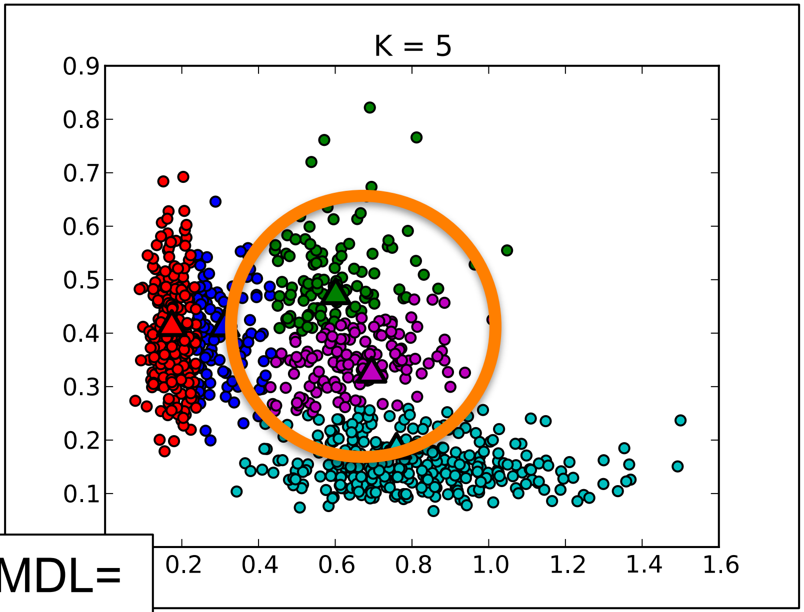
Ground Truth Mixture Data

Agglomerative Clustering with MDL



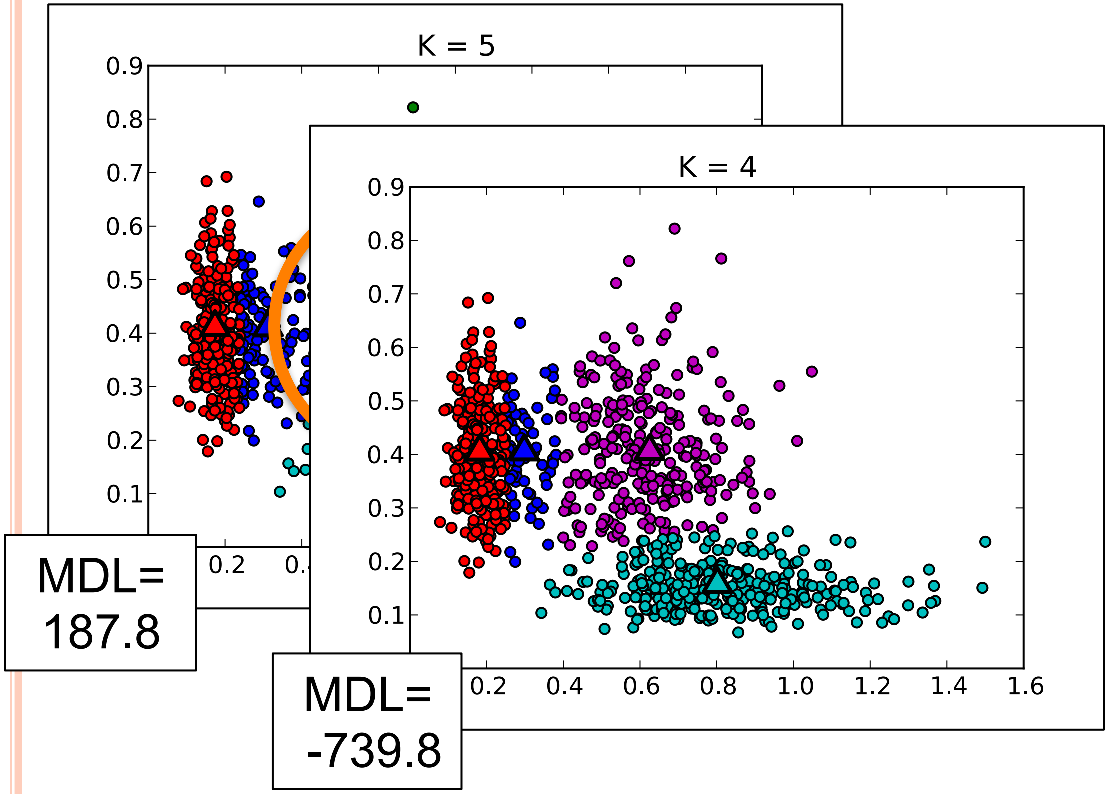
MDL=
187.8

Agglomerative Clustering with MDL

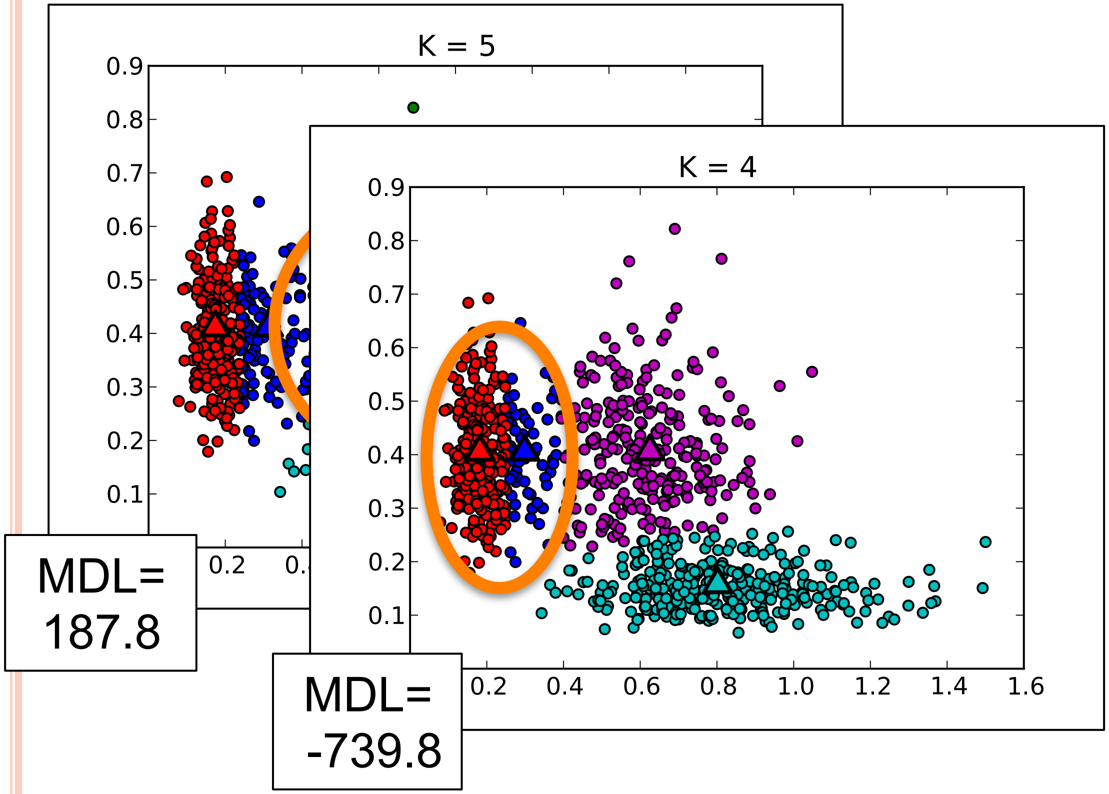


MDL=
187.8

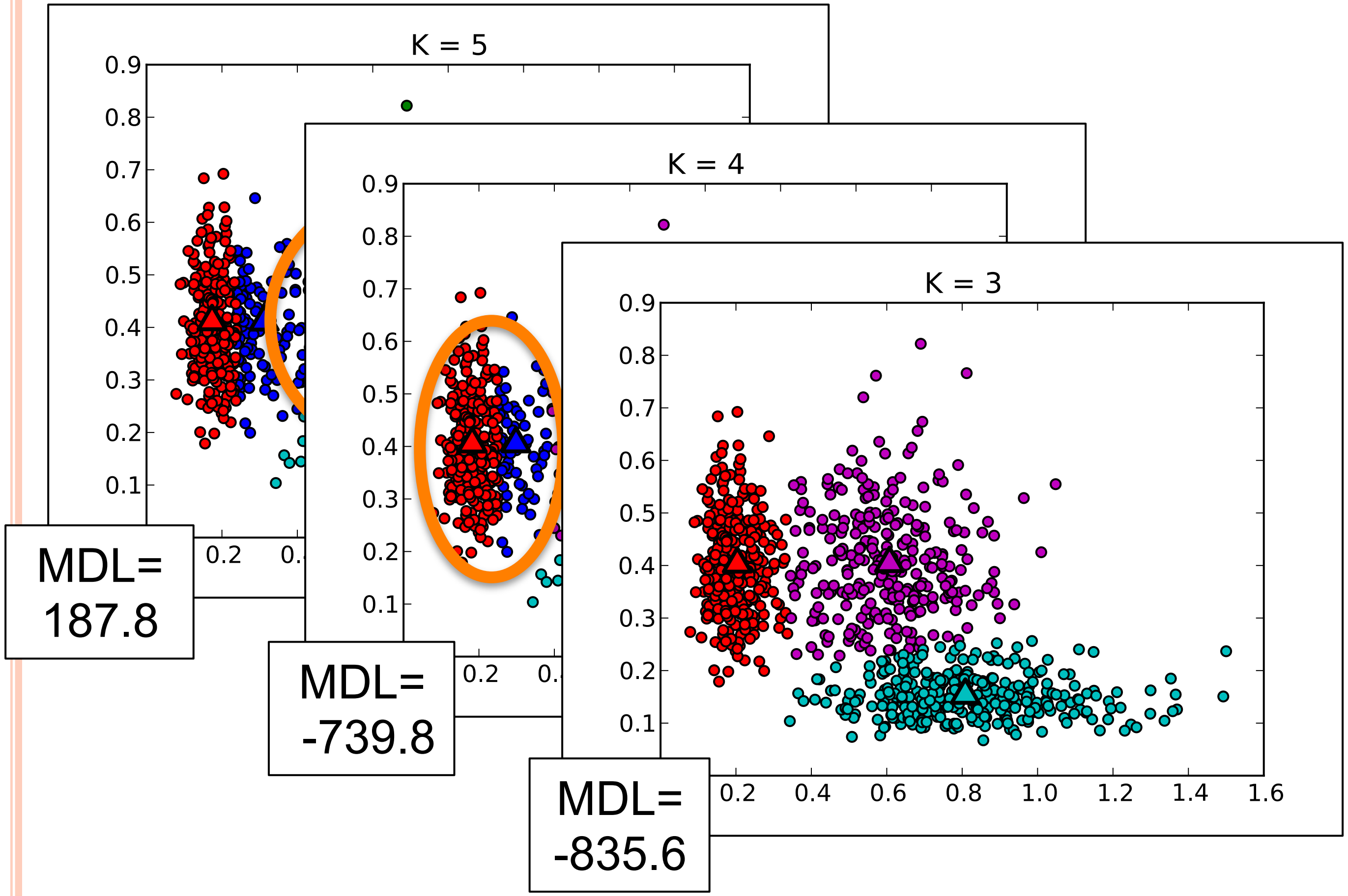
Agglomerative Clustering with MDL



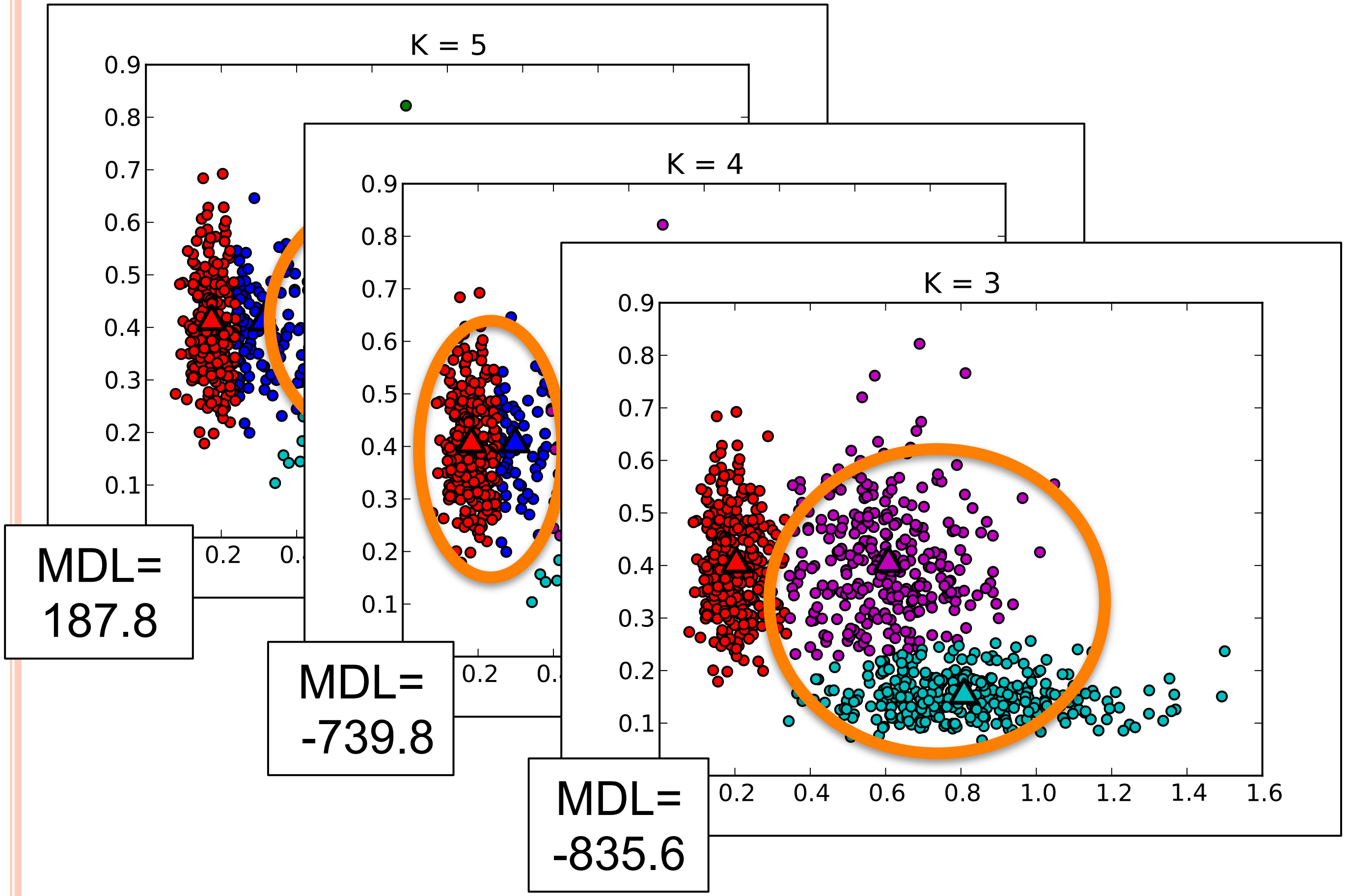
Agglomerative Clustering with MDL



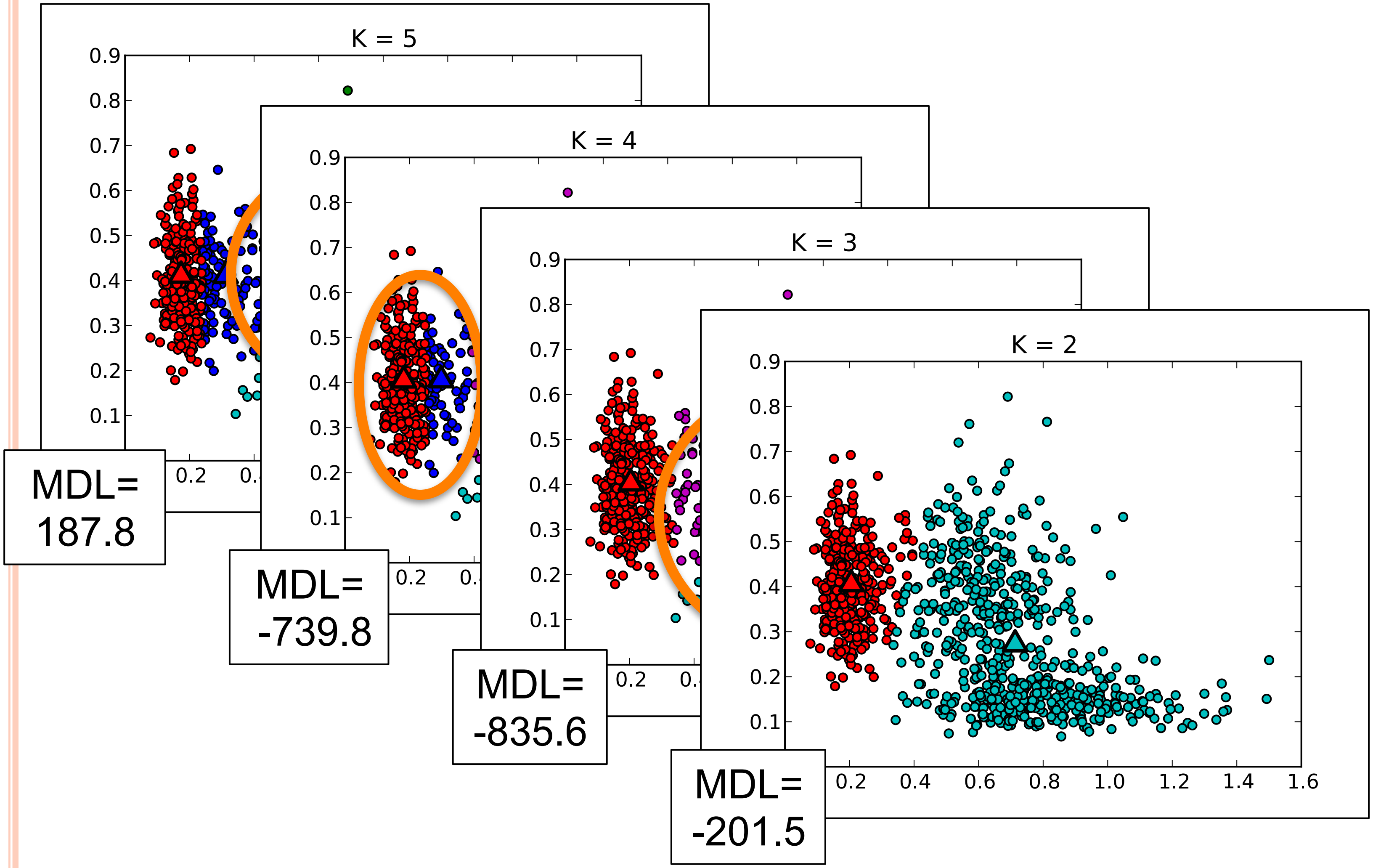
AGGLOMERATIVE CLUSTERING WITH MDL



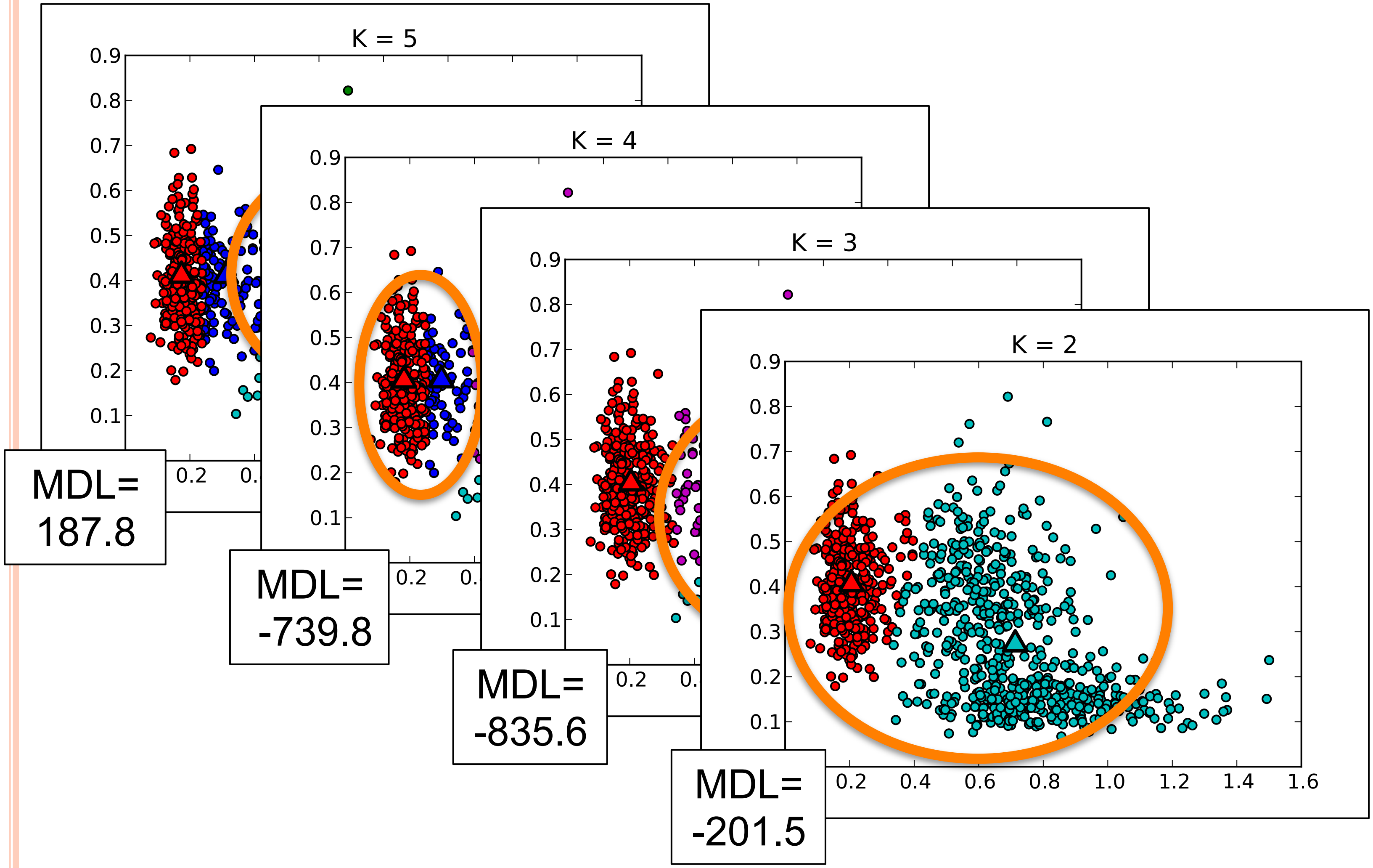
Agglomerative Clustering with MDL



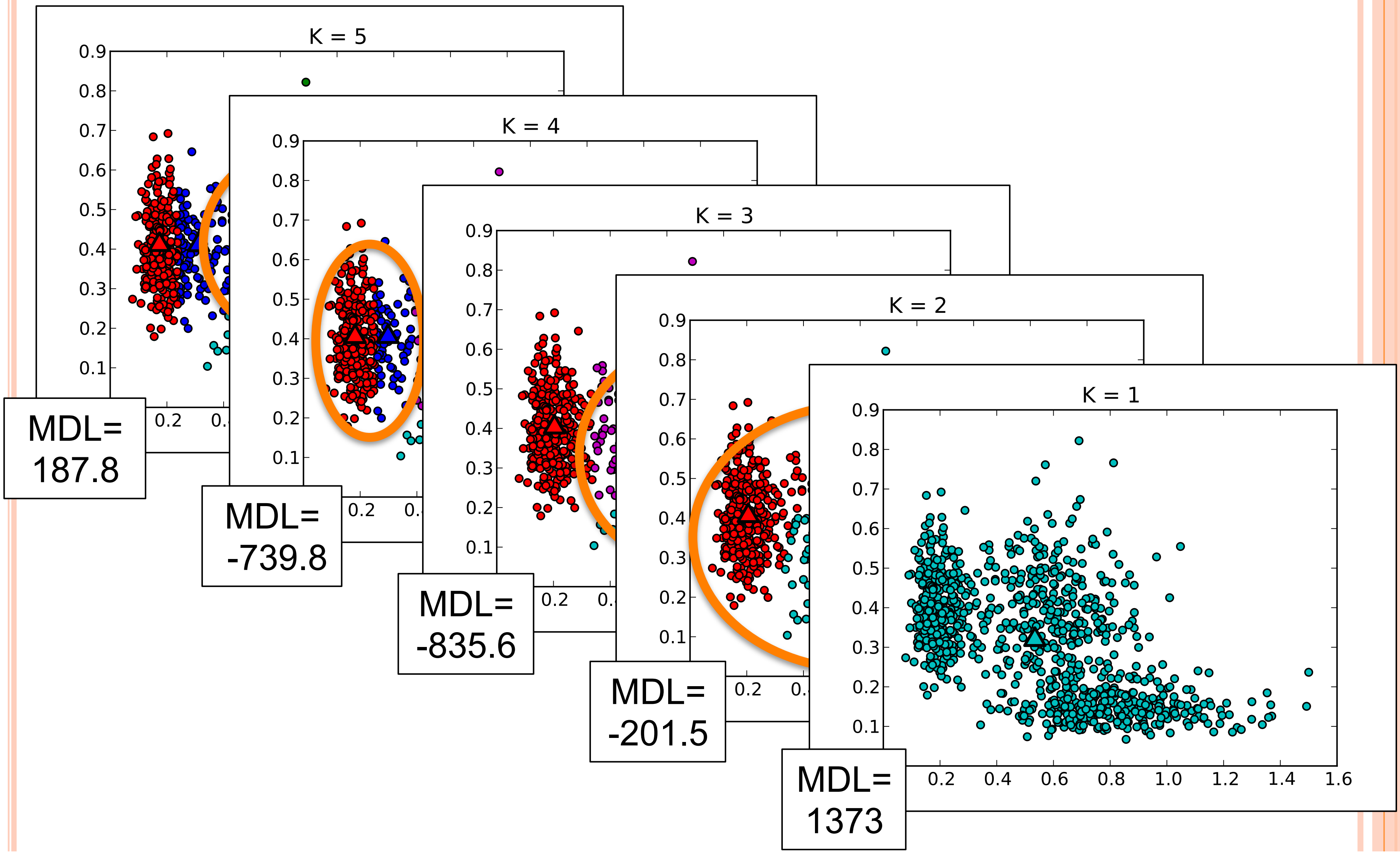
Agglomerative Clustering with MDL



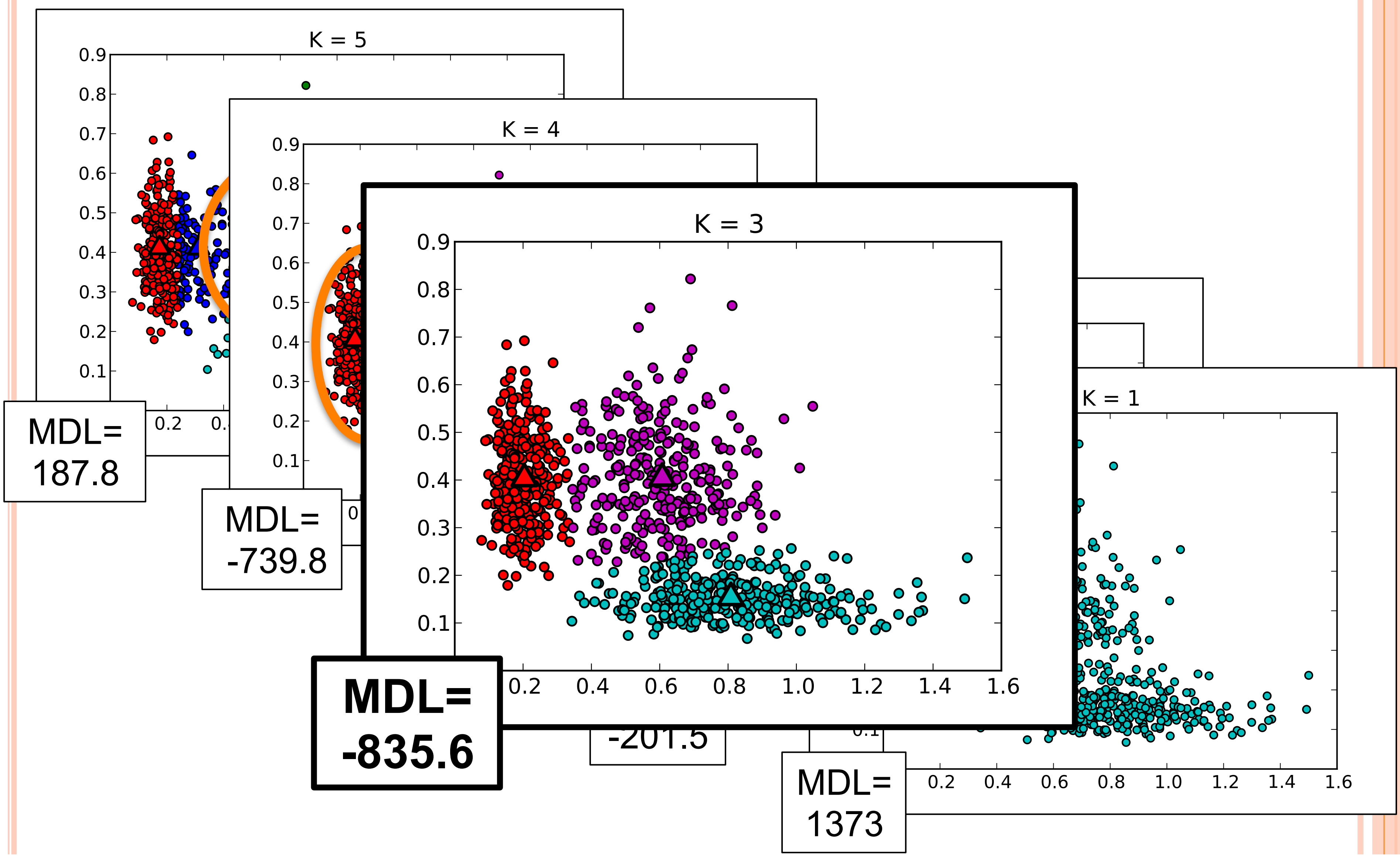
AGGLOMERATIVE CLUSTERING WITH MDL



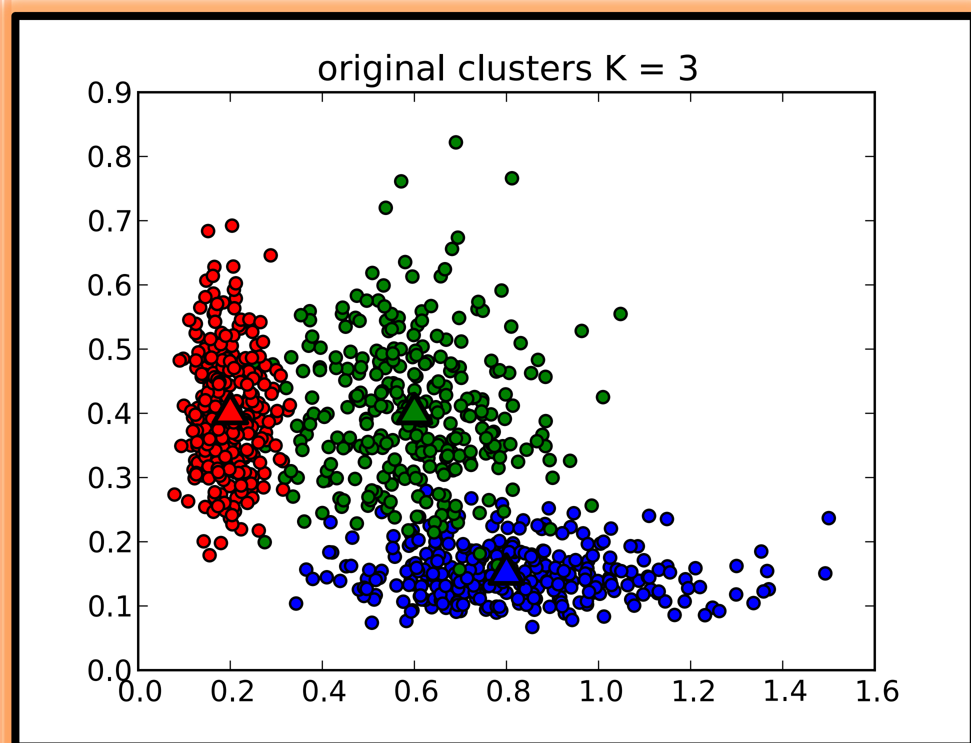
Agglomerative Clustering with MDL

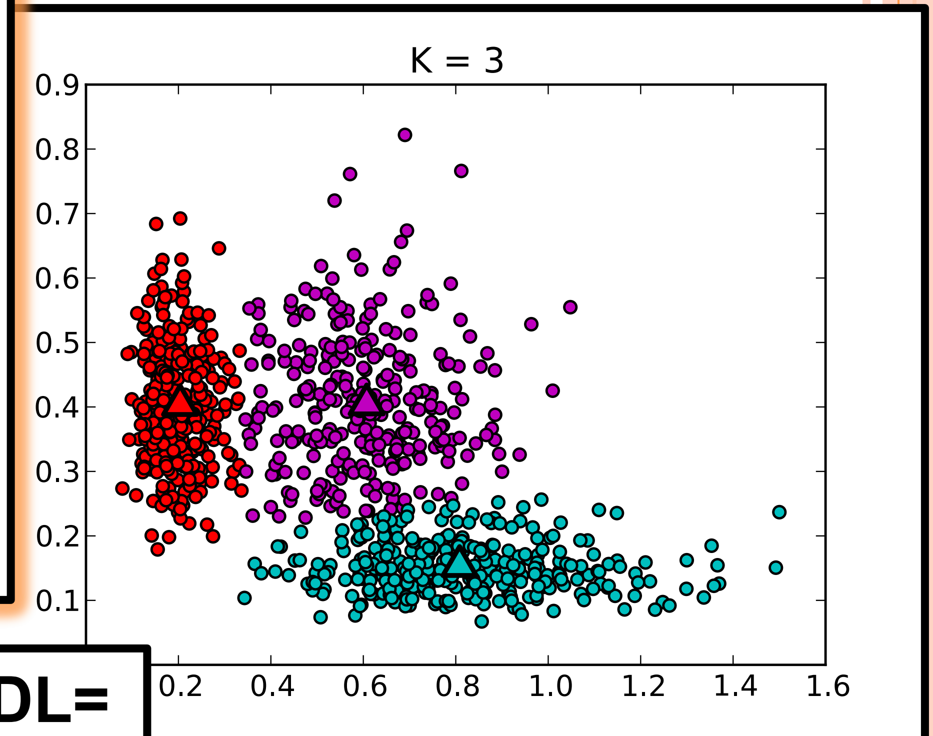


AGGLOMERATIVE CLUSTERING WITH MDL



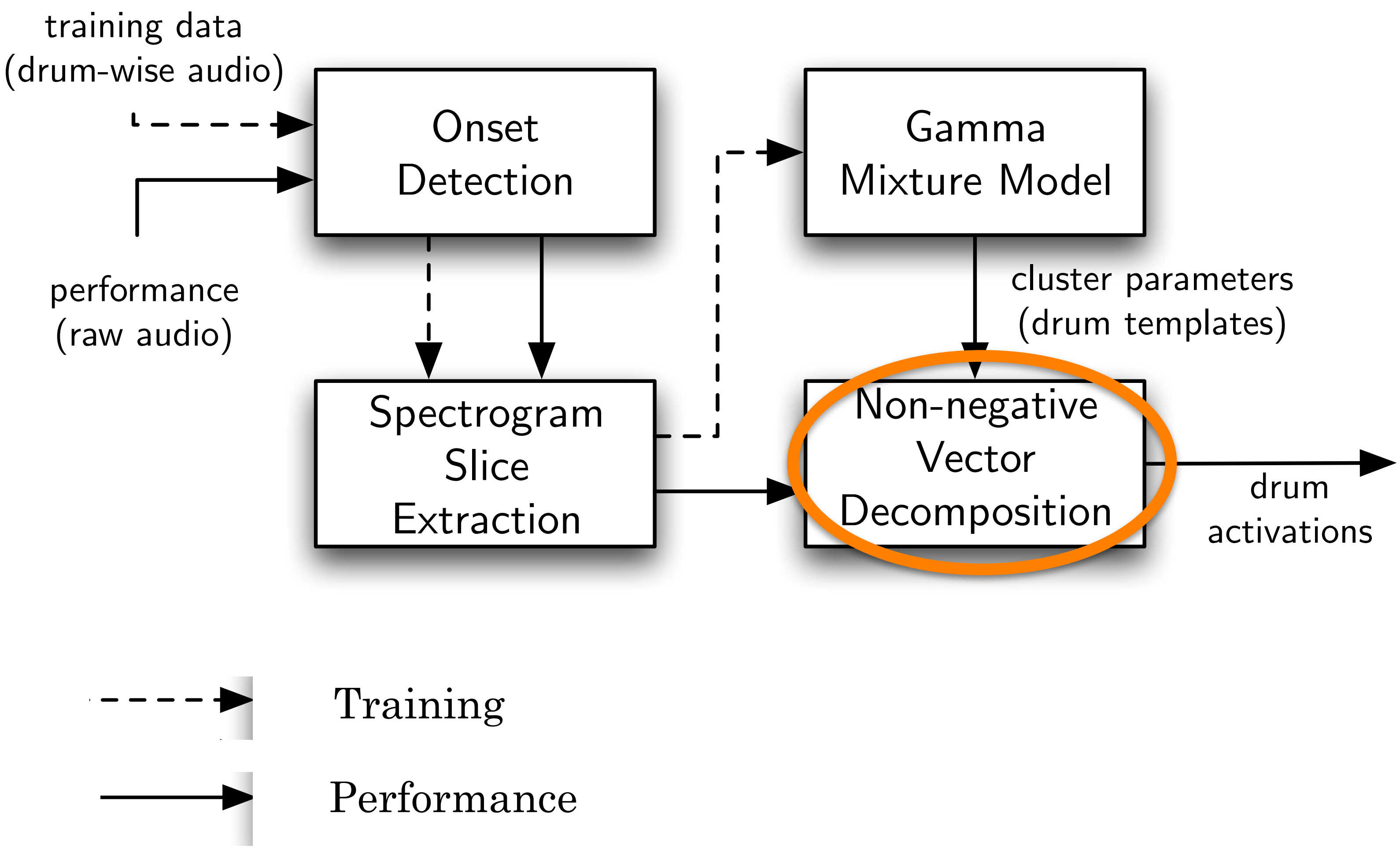
Agglomerative Clustering with MDL





MDL=
-835.6

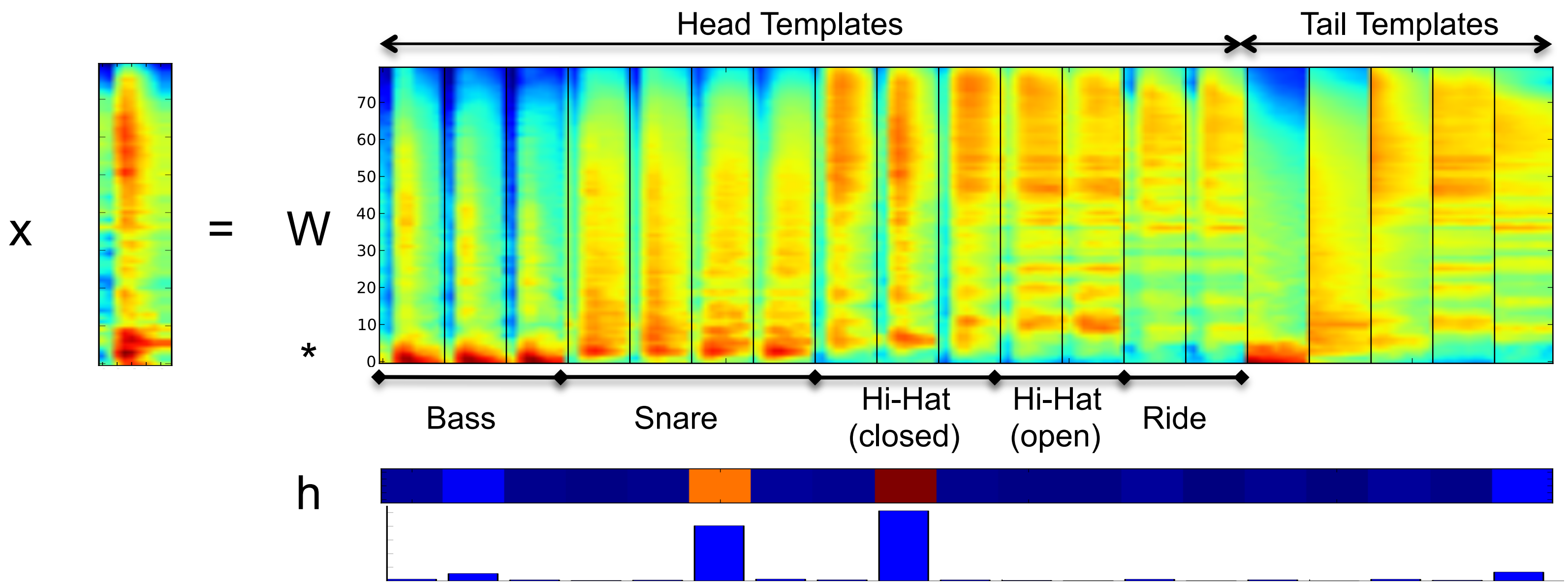
Drum Separation System



Decomposing Onsets onto Templates

- Non-negative Vector Decomposition (NVD)
  - A simplification of Non-negative Matrix Factorization (NMF)
  - W matrix contains drum templates in its columns.

$$\min_{\vec{h}} d_{IS}(\vec{x}, W\vec{h}), \quad h_i \geq 0 \quad \forall i$$



Decomposing Onsets onto Templates

- To solve this problem:

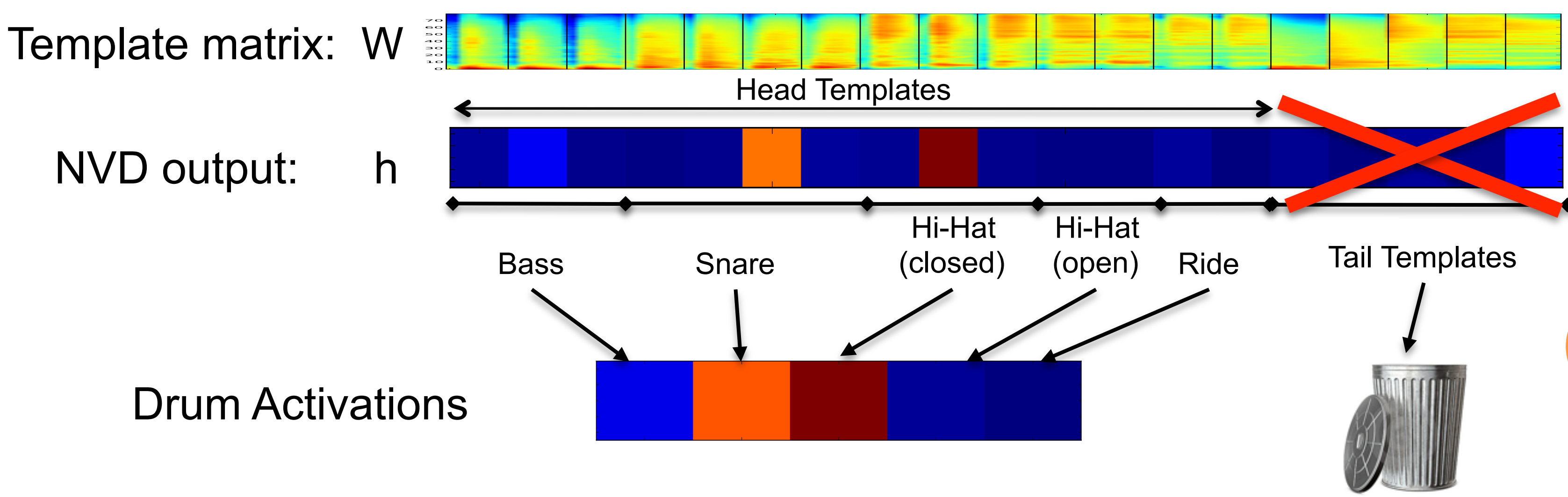
$$\min_{\vec{h}} d_{IS}(\vec{x}, W\vec{h}), \quad h_i \geq 0 \ \ \forall i$$

- We use the IS distance as the cost function in the above.
  - While the IS distance is not strictly convex, in practice it is non-increasing under the following update rule:

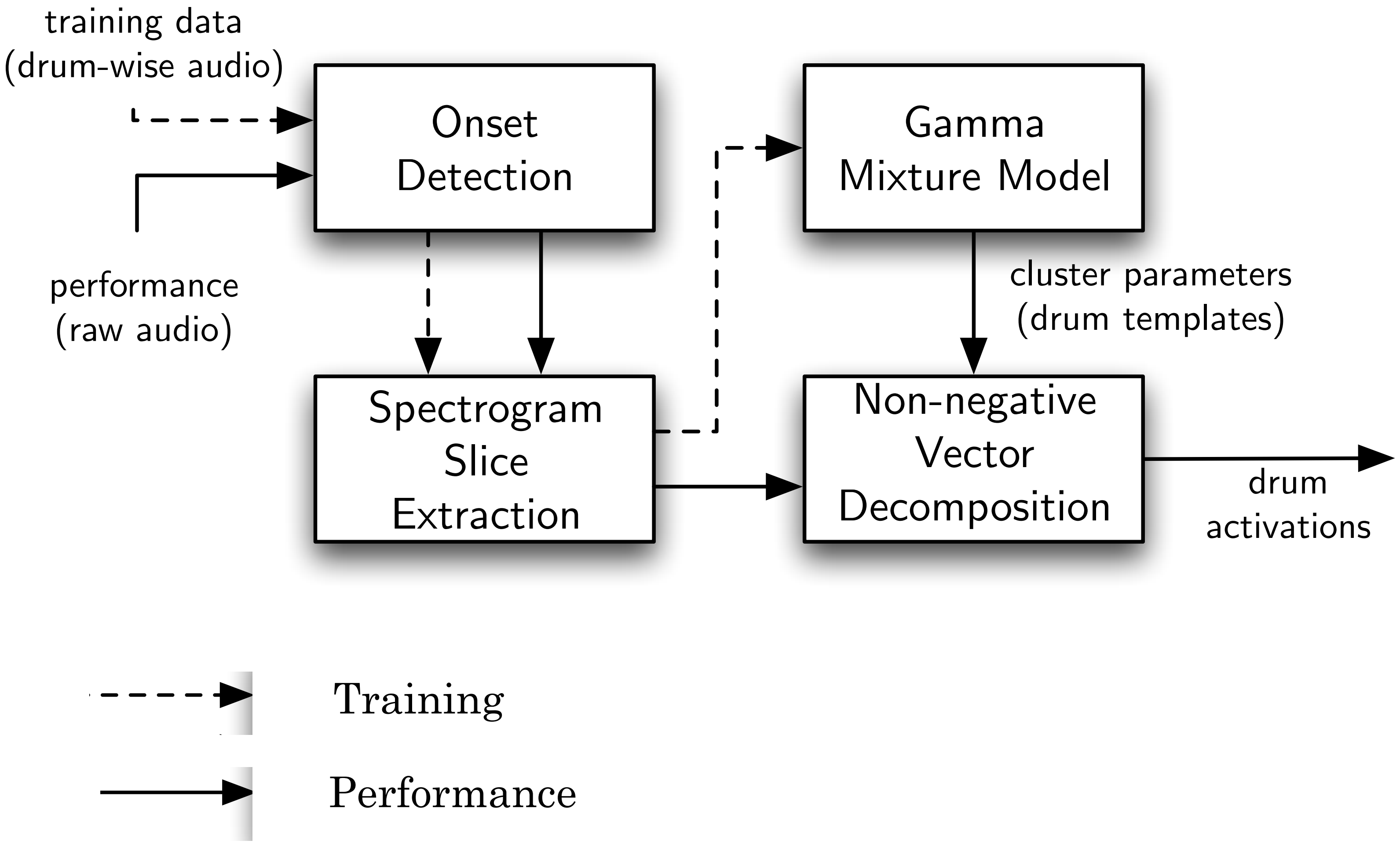
$$\vec{h}_i \leftarrow \vec{h}_i . \frac{W^T((W\vec{h}_i)^{.-2} . \vec{x}_i)}{W^T(W\vec{h}_i)^{.-1}}$$

DECOMPOSING ONSETS ONTO TEMPLATES

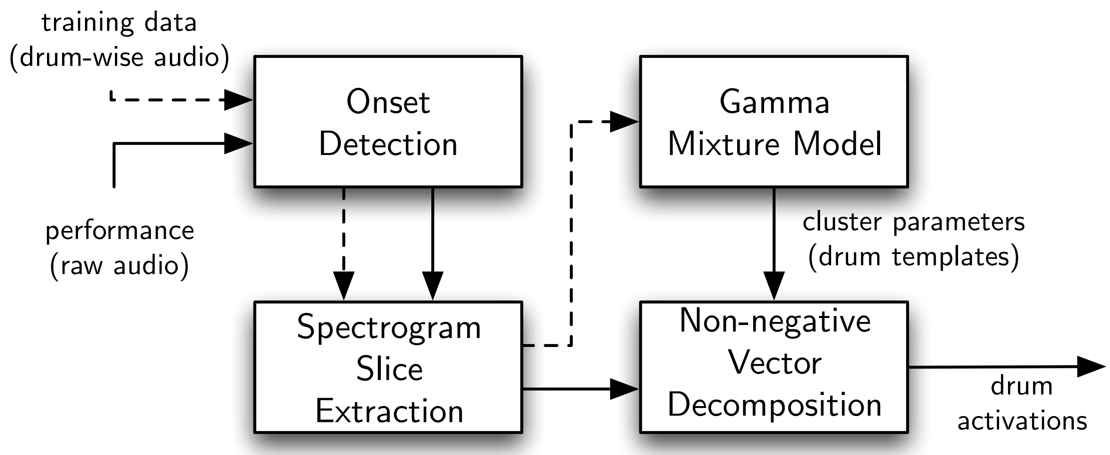
- What do we do with the output of NVD?
  - The **head** template activations for a single drum are **summed** to get the total activation of that drum.
  - The **tail** template activations are **discarded**.
    - They simply serve as "decoys" so that the long decay of a previous onset does not affect the current decomposition as drastically.



Drum Separation System



Building/Testing the System

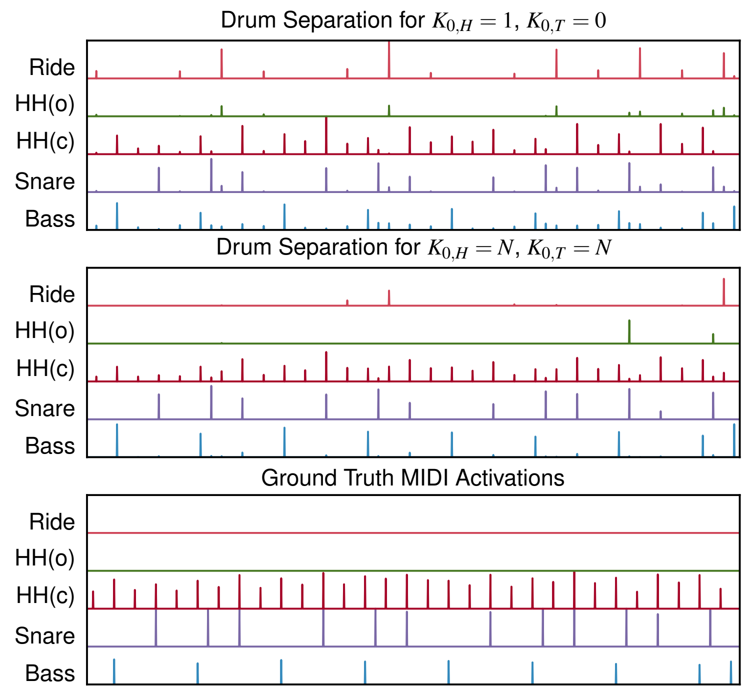


- Implemented in Python with Scipy
  - NVD can easily be done in real-time (100ms latency)
  - Agglomerative Gamma Mixture Model training takes ~20 seconds for 5 drums.
    - *Could be reduced to < 1 sec using a GPU implementation.*
- Parameters to vary for testing:
  - Number head/tail templates per drum
    - {0, 1, MDL-optimal}

Quantitative Results

- We test using a total of 10 drum performances:
  - 10 minutes total, 2922 drum onsets
  - Recorded as midi data
    - Roland V-Drums
  - Audio created using multi-sampled drum kit
    - Superior Drummer 2.0
- Onset detection results
  - **85% recall, 99.9% precision**
- Decomposition results
  - **Cosine similarity for true activations**
  - **Amplitude sum for false activations**

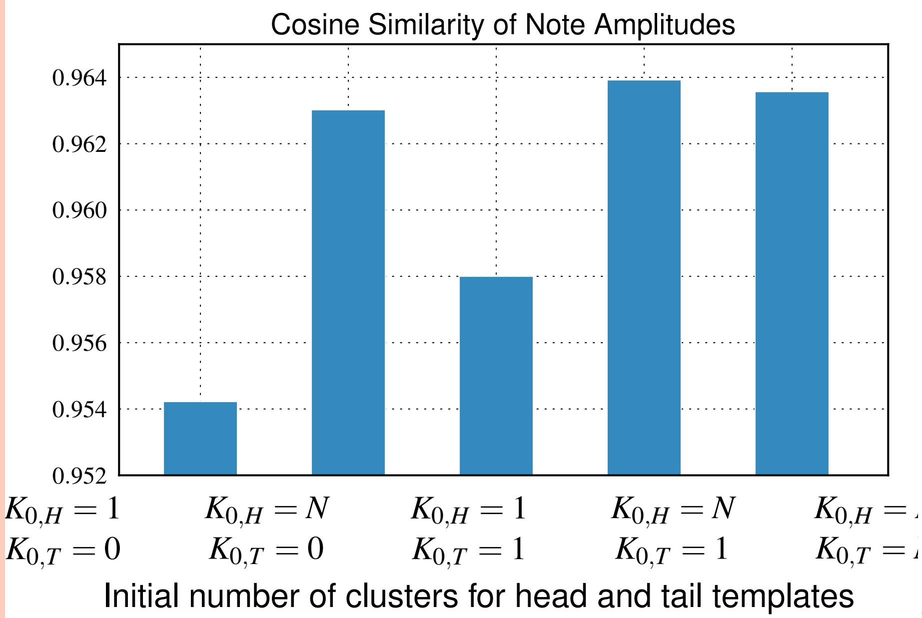
Example Activations



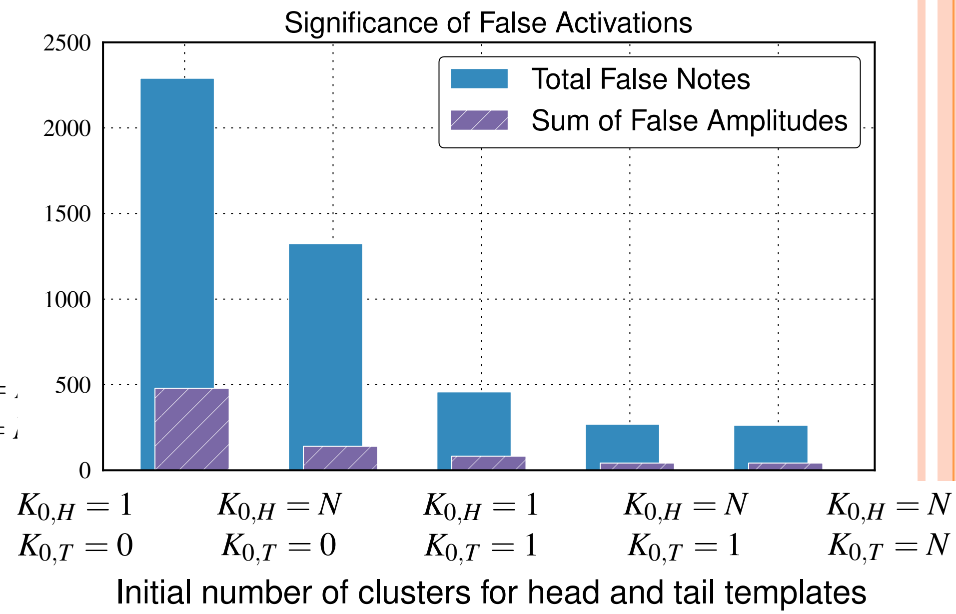
$$S_{\cos}(\vec{x},\vec{y}) = \frac{\vec{x}\cdot\vec{y}}{||\vec{x}||\,||\vec{y}||}$$

Quantitative Results

Cosine similarity of True Activations



Amplitude sum of False Activations



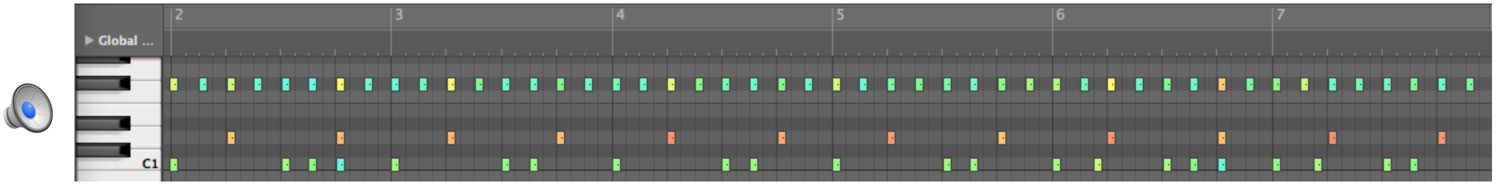
- Significant improvements seen with:
  - > 1 head template
  - 1 tail templates

AUDIO EXAMPLES

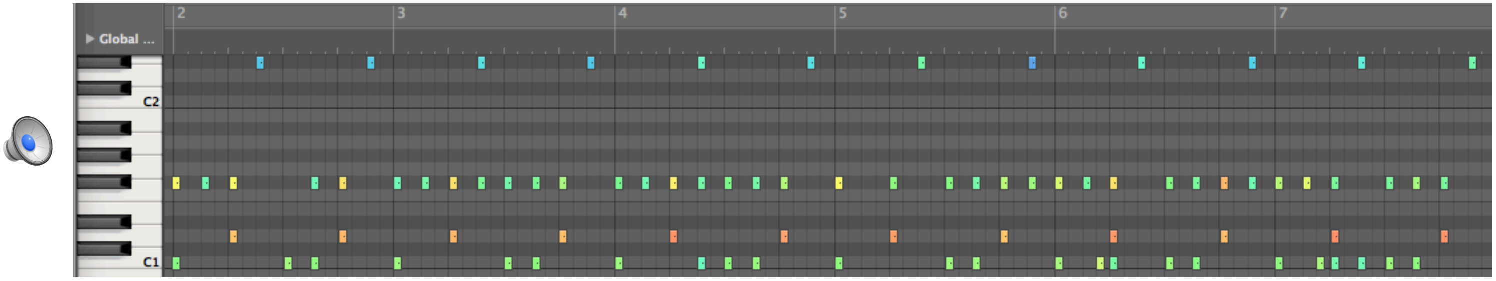
- Track 1 - Basic 4/4 rock beat (quantized)

Original Performance

KH=MDL-Optimal, KT=1



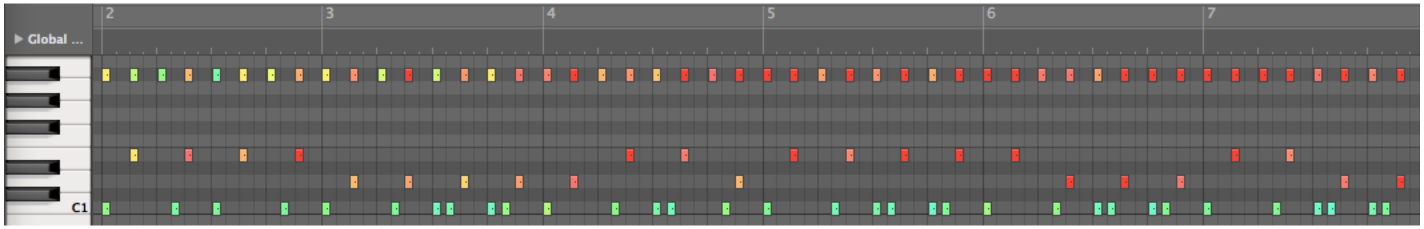
KH=1, KT=0



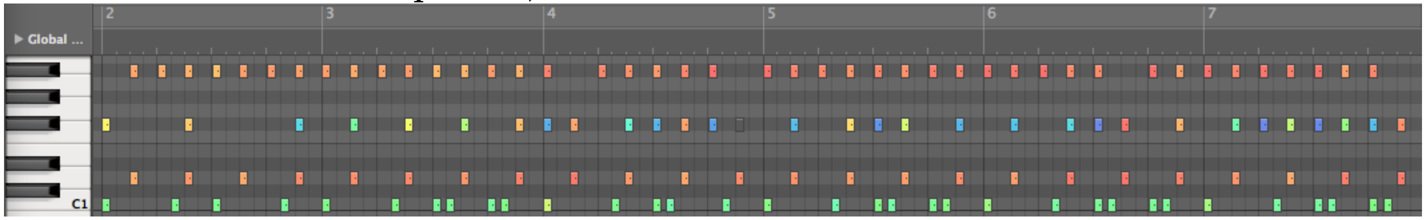
Audio Examples

- Track 3 - Cut time rock with open hi-hat

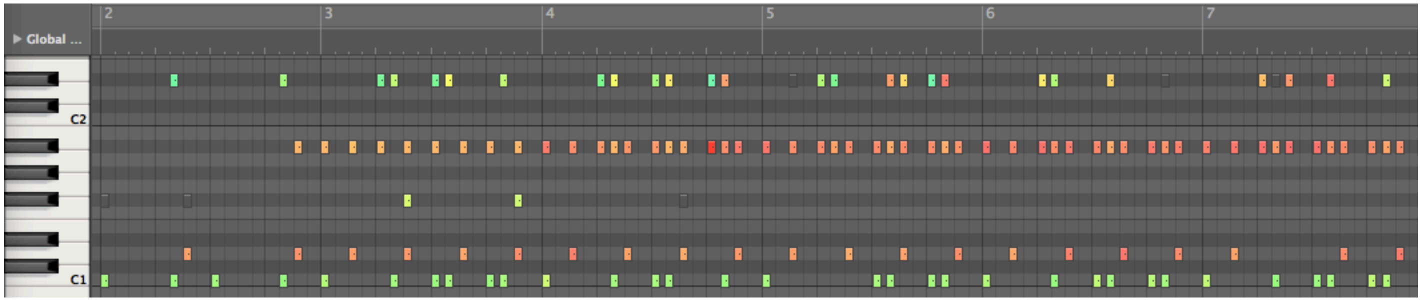
Original Performance



KH=MDL-Optimal, KT=1



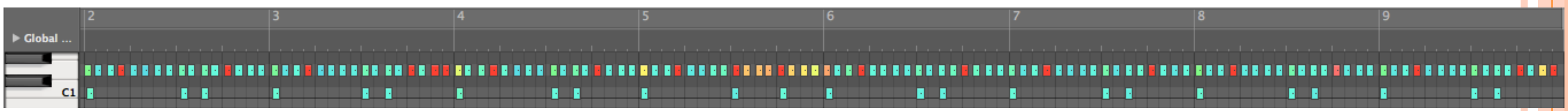
KH=1, KT=0



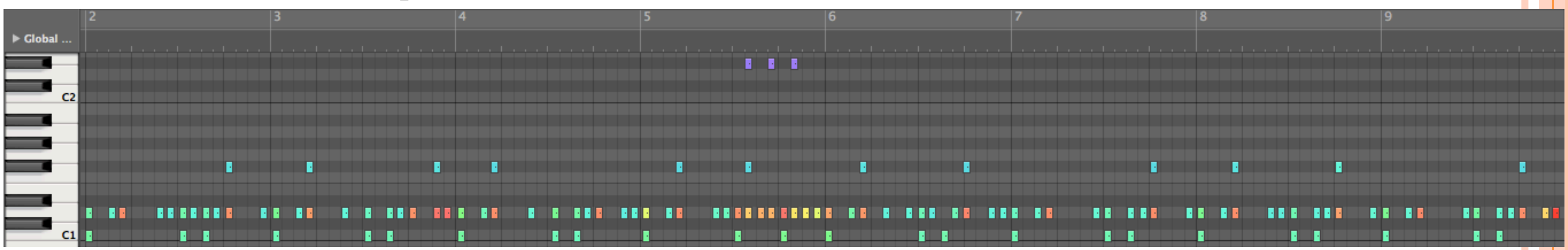
AUDIO EXAMPLES

- Track 7 - Accented snare drum roll.

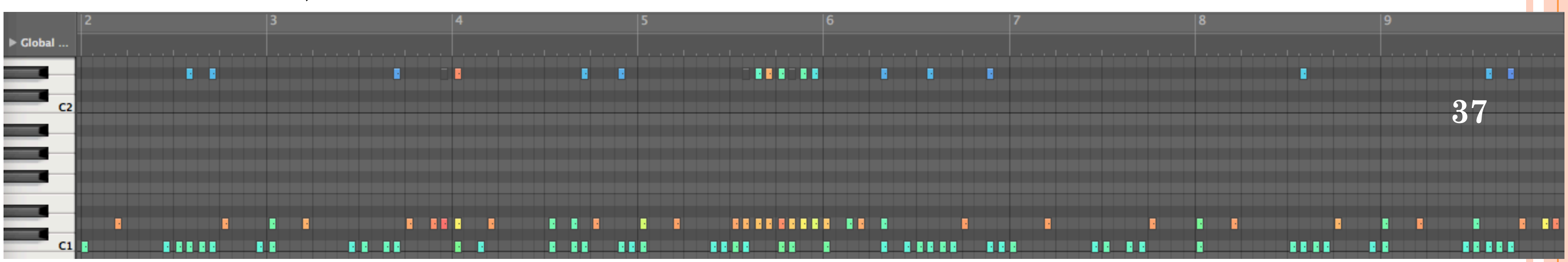
Original Performance



KH=MDL-Optimal, KT=1



KH=1, KT=0



SUMMARY

- Drum separation front end for a complete drum understanding system.
- Gamma Mixture Model
  - Cheaper to train than GMM (no covariance matrix)
  - More stable than GMM (no covariance matrix)
  - Allows clustering with perceptual Itakura-Saito distance
- Non-negative Vector Decomposition
  - Greatly improved with tail templates and multiple head templates per drum.
- Next steps
  - Explore online training of templates.
  - Integration with complete drum understanding system.

KIITOS


"MY DW DRUMS ARE A UNIVERSE OF SOUND"
- TERRY BOZZIO
WWW.DWDRUMS.COM

Extra Slides

Gamma Mixture Model

- Multivariate Gamma (independent components):

$$\mathrm{p}(\vec{y}|\vec{\lambda},k) = \prod_{i=1}^{M} \frac{\lambda_i^k y_i^{k-1} e^{-\lambda_i y_i}}{\Gamma(k)}$$

- Mixture density:

$$\mathrm{p}(\vec{y}_n|\theta) = \sum_{l=1}^{K} \pi_l \mathrm{p}(\vec{y}_n|\vec{\lambda}_l,k) \qquad \theta = \{\vec{\lambda}_l, \pi_l\}_{l=1}^{K}$$

$$\pi_l = \mathrm{p}(x_n = l)$$

THE EM ALGORITHM: GAMMA EDITION

- E-step: (compute posteriors)

$$p(x_n = l|\vec{y}_n, \theta^{(t)}) = \frac{\pi_l \exp\left(-k\, d_{\mathrm{IS}}(\vec{y}_n, \vec{\mu}_l)\right)}{\sum_{j=1}^{K} \pi_j \exp\left(-k\, d_{\mathrm{IS}}(\vec{y}_n, \vec{\mu}_j)\right)}$$

- M-step: (update parameters)

$$\begin{aligned} N_l^* &= \sum_{n=1}^{N} p(x_n = l|\vec{y}_n, \theta^{(t)}) \\ \vec{\lambda}_l &\leftarrow \frac{kN_l^*}{\sum_{n=1}^{N} \vec{y}_n p(x_n = l|\vec{y}_n, \theta^{(t)})} \\ \pi_l &\leftarrow \frac{N_l^*}{N} \end{aligned}$$

AGGLOMERATIVE CLUSTERING

- *How many clusters to train?*
- We use Minimum Description Length (MDL) to choose the number of clusters.
  - Negative log-likelihood
  - + penalty term for number of clusters.

$$
\begin{aligned}
\mathrm{MDL}(K,\theta) &= -\sum_{n=1}^{N} \log\left(\sum_{l=1}^{K} \mathrm{p}(\vec{y}_n|\vec{\lambda}_l)\pi_l\right) + \frac{1}{2}L\log(NM) \\
L &= KM + (K-1)
\end{aligned}
$$

  - 1. Run EM to convergence.
  - 2. Merge the two most similar clusters.
  - 3. Repeat 1,2 until we have a single cluster.
  - 4. *Choose parameter set with smallest MDL.*