Support Vector Machine Active Learning for Music Retrieval

Michael I. Mandel, Graham E. Poliner, Daniel P.W. Ellis

Mansoor Siddiqui
Jan. 29, 2009

Problem Overview

- User presents classifier with some “seed” songs
- The classifier's job is to present the user with similar songs

Passive vs. Active Learning

- **Passive Learning:**
  - Classifier is trained on large pool of randomly selected labeled data without user involvement
- **Active Learning:**
  - Classifier asks user to label only those instances that would be most informative (this is called *"relevance feedback"*)
  - This approach is more robust to user subjectivity and we don't need a massive pre-labeled data set

Algorithm Overview

1. Seed the search with representative song(s).
2. Acquire initial negative examples by e.g. presenting randomly selected songs for labeling
3. Train an SVM on all labeled examples
4. Present the user with the most relevant songs (those with the greatest positive distance to the decision boundary)
5. If the user wishes to refine the search further, present the most informative songs (those closest to the decision boundary) for labeling and repeat 3-5.

Support Vector Machines

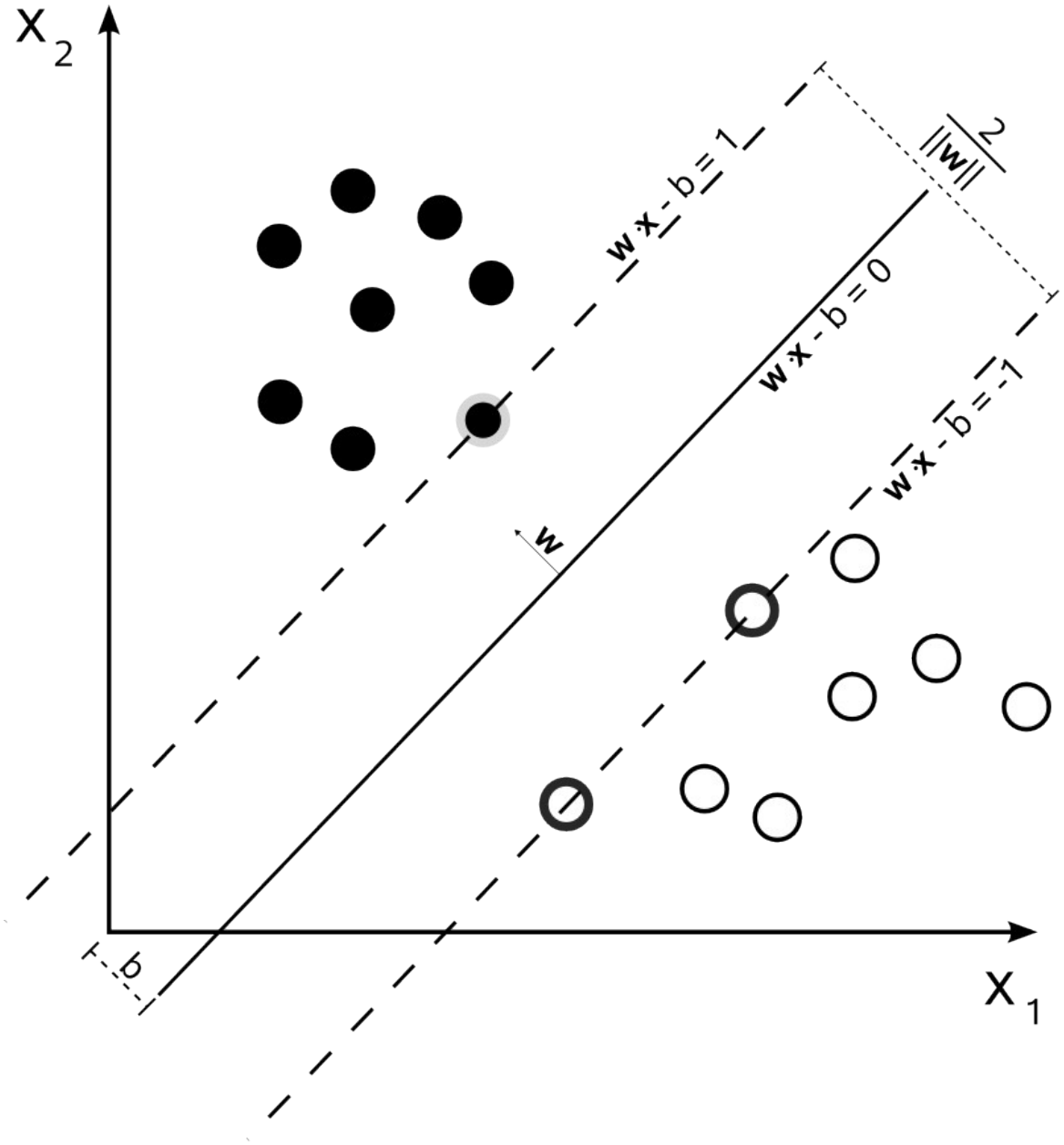
- Given:
  - Data points $\{\mathbf{X}_0, \ldots \mathbf{X}_N\}$
  - Class labels $\{y_0, \ldots y_N\}$, $y_i \in \{-1, 1\}$
- We want to separate the two classes by a hyperplane:

$$y_i(\mathbf{w}^T \mathbf{X}_i + b) > 0 \; \forall i$$

- ...such that the margin between the two classes is maximized:

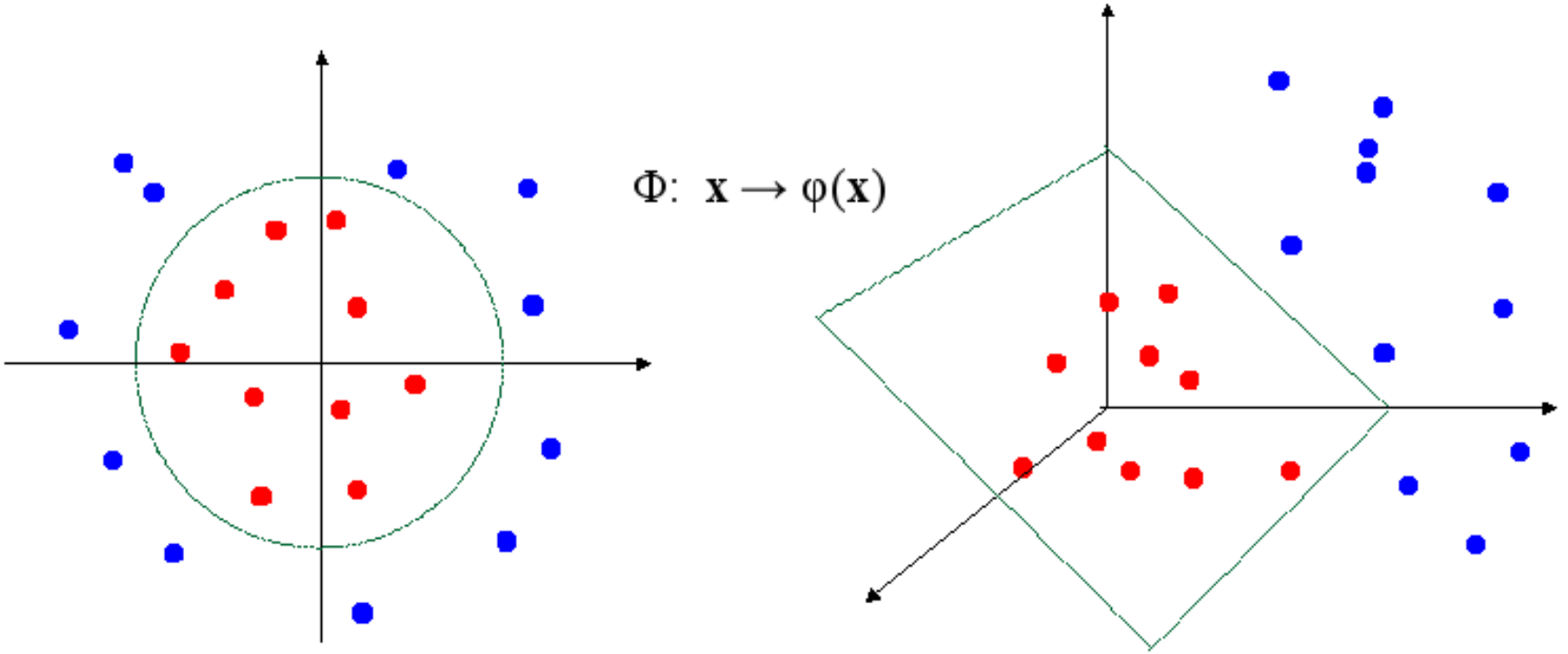
$$\mathbf{w} = \sum_{i=0}^{N} \alpha_i y_i \mathbf{X}_i$$

Support Vector Machines



The Kernel Trick

- For data that is not linearly separable, we can map the data to a higher dimensional feature space in which it is linearly separable:



The Kernel Trick

- The radial basis function (RBF) kernel was used:

$$K(\mathbf{X}_i, \mathbf{X}_j) = e^{-\gamma D^2(\mathbf{X}_i, \mathbf{X}_j)}$$

- *D* is any distance function

Active Learning: Parameter Space

- Earlier we saw this equation defining a hyperplane that separates the data points:

$$y_i(\mathbf{w}^T\mathbf{X}_i + b) > 0 \; \forall i$$

- Instead of thinking of **X** as the data points and **w** as the normal to the hyperplane, we can think of **X** as the normal and **w** as the data points
- This interpretation is called *"parameter space"*

Active Learning: Parameter Space

- In parameter space, the set of all possible **w** (referred to as *"version space"*) are points that are bounded by the hyperplanes formed by **X**
- We want to find the **w** in version space that defines the maximum margin

Active Learning: Parameter Space

- Whenever we have a new labeled **X**, the version space shrinks (due to being further constrained)
- So the fastest way to shrink the version space is by asking the user to label new **X** values that would split the version space in half

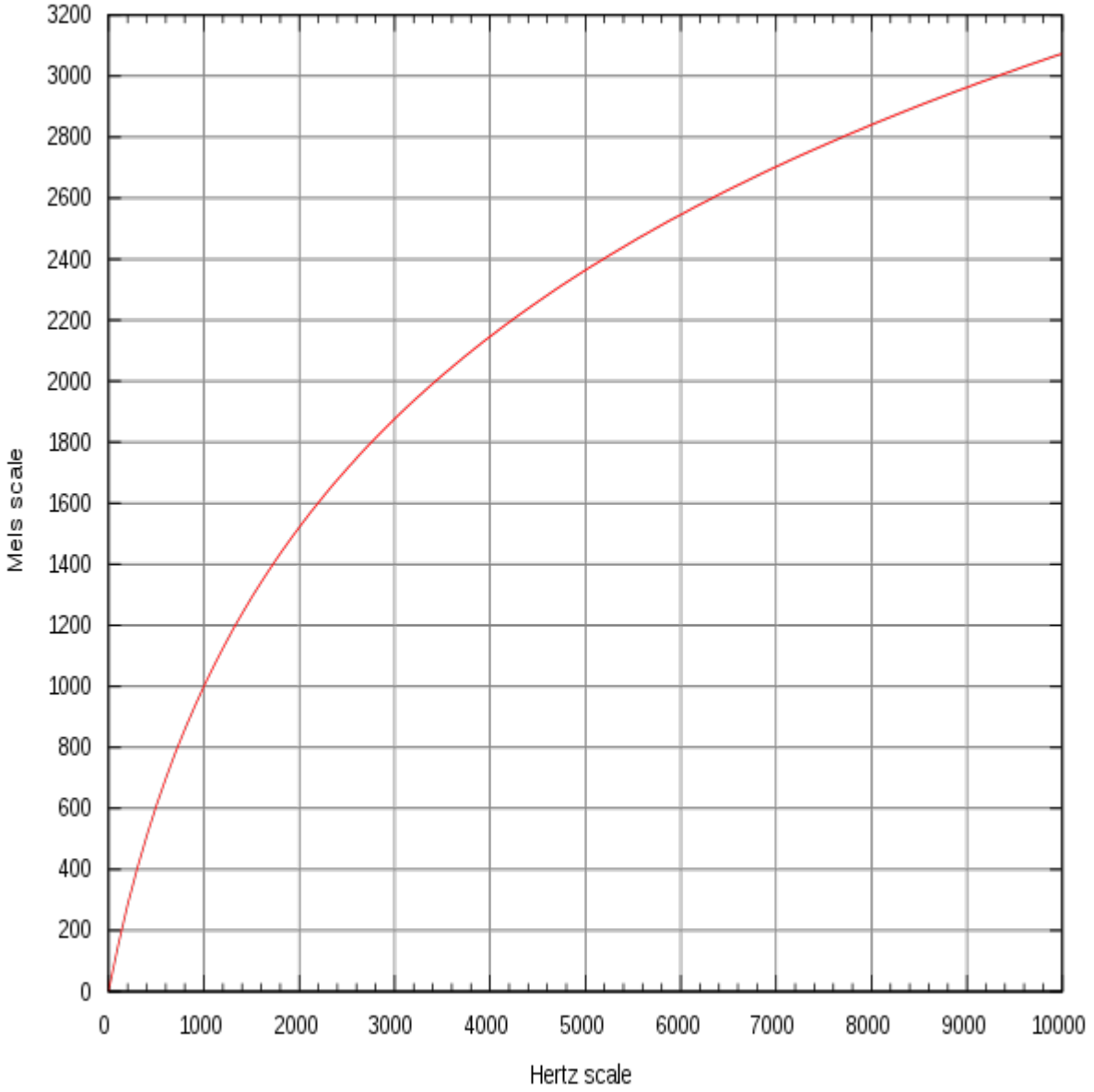
Choosing Informative Points

- When asking the user to label points **X**, we can take several approaches to determine which points would be most informative:
  - Take points closest to the center of the version space (i.e. points closest to the decision boundary)
  - Use *"angle diversity"*, which balances closeness to decision boundary with coverage of feature space
  - Randomly select songs **X** for user to label

Feature Set: MFCCs

- Break signal into overlapping frames of ~25ms (short enough to assume the signal is stationary)
- Take discrete Fourier transform of each window
- Take log-magnitude of the result
- Warp result to the Mel frequency scale
- Then take the inverse discrete cosine transform
- Mahalanobis distance measure used

Feature Set: MFCCs



Feature Set: GMMs

1) Song's MFCCs are fit to a single Gaussian
   - Songs are parameterized by mean and covariance

2) Song's MFCCs are fit to a mixture model of 20 Gaussians

- Kullback-Leibler (KL) divergence used as a distance measure

Feature Set: Anchor Posteriors

- The entire song set is modeled using a GMM
- The posterior probabilities that the song belongs to each of the Gaussians in the GMM are calculated:

$$P(k|\mathbf{X}) \propto p(\mathbf{X}|k)P(k) = P(k)\prod_{t=1}^{T} p(\mathbf{x}_t|k)$$

$$f(k) = P(k)\prod_{t=1}^{T} p(\mathbf{x}_t|k)^{1/T} \propto \prod_{t=1}^{T} p(k|\mathbf{x}_t)^{1/T}$$

- Euclidean distance used

Feature Set: Fisher Kernel

- The entire song set is modeled using a GMM
- Describes each song by partial derivatives of the log-likelihood of the song with respect to each Gaussian mean:

$$\nabla_{\mu_k} \log P(\mathbf{X} \mid \mu_k) = \sum_{t=1}^{T} P(k \mid \mathbf{x}_t) \Sigma_k^{-1} (\mathbf{x}_t - \mu_k)$$

- Feature vector of size 650 (50 means x 13 dimensions)

Feature Set: Fisher Kernel

- The Fisher kernel is essentially a gradient (partial derivatives with respect to change in the means of each Gaussian)
- A more compact feature is the magnitude of the gradient, which is only 50 dimensional

Feature Set: Summary

GMM over	Feature	Parameters	Representation	Distance measure $D^2(\mathbf{X}_i, \mathbf{X}_j)$
	MFCC Stats	104	$[\mu^T \ \mathrm{vec}(\Sigma)^T]$	$(\mu_i-\mu_j)^T\Sigma_\mu^{-1}(\mu_i-\mu_j)+\mathrm{vec}(\Sigma_i-\Sigma_j)^T\Sigma_\Sigma^{-1}\,\mathrm{vec}(\Sigma_i-\Sigma_j)$
Song	KL 1G	104	μ, Σ	$tr(\Sigma_i^{-1}\Sigma_j+\Sigma_j^{-1}\Sigma_i)+(\mu_i-\mu_j)^T(\Sigma_i^{-1}+\Sigma_j^{-1})(\mu_i-\mu_j)-2d$
	KL 20G	520	$\{\mu_k, \Sigma_k\}_{k=1\ldots20}$	$\frac{1}{N}\sum_{n=1}^{N}\log\frac{p_i(\mathbf{x}_{ni})}{p_j(\mathbf{x}_{ni})}+\frac{1}{N}\sum_{n=1}^{N}\log\frac{p_j(\mathbf{x}_{nj})}{p_i(\mathbf{x}_{nj})}$
	GMM Posteriors	100	$\{\frac{1}{T}\sum_{t=1}^{T}\log p(k\mid\mathbf{x}_t)\}_{k=1\ldots50}$	$\sum_{k=1}^{50}\log^2\frac{p(X_i\mid k)^{1/T_i}}{p(X_j\mid k)^{1/T_j}}$
Corpus	Fisher	650	$\{\nabla_{\mu_k}\}_{k=1\ldots50}$	$\sum_{k=1}^{50}\left[\nabla_{\mu_k}\log p(\mathbf{X}_i\mid\mu_k)-\nabla_{\mu_k}\log p(\mathbf{X}_j\mid\mu_k)\right]^2$
	Fisher Mag	50	$\{\lvert\nabla_{\mu_k}\rvert\}_{k=1\ldots50}$	$\sum_{k=1}^{50}\left[\lvert\nabla_{\mu_k}\log p(\mathbf{X}_i\mid\mu_k)\rvert-\lvert\nabla_{\mu_k}\log p(\mathbf{X}_j\mid\mu_k)\rvert\right]^2$

The Data Set

- 1210 pop songs
- 18 artists
- 90 albums
- Each artist has at least 5 albums (3 for training, 2 for testing)
- Each album has at least 8 tracks
- We are interested in artist, mood, and style classification

The Data Set: Mood and Style

- Labels for mood and style obtained from AMG
- *"Moods"* are adjectives describing the feel of a song (e.g. "cerebral", "hypnotic", "silly")
- *"Styles"* are sub-genre categories (e.g. "pop-punk", "prog-rock/art rock", "speed metal")

Mood	Songs	Style	Songs
Rousing	527	Pop/Rock	730
Energetic	387	Album Rock	466
Playful	381	Hard Rock	323
Fun	378	Adult Contemporary	246
Passionate	364	Rock & Roll	226

The Data Set: Mood and Style

- Only moods and styles that appeared in at least 50 songs were used:
  - 32 styles
  - 100 moods
- **Assumption:** Moods and styles are given by AMG only to artists and albums; it is assumed that mood and style labels apply to all songs for the relevant artist/ablum

Experiment #1: Finding the Optimal Feature Set

- Artist, mood, and style classification accuracy measured for each of the six feature sets
- Passive learning used
- “Precision-at-20” evaluation
  - For moods and styles, success is measured only on top 20 returned songs

Experiment #1: Results

Feature	Accuracy Artist 18-way	Precision-at-20 Mood ID	Style ID
MFCC Stats	.682	.497	.755
Fisher Kernel	.543	.508	.694
KL 1G	.640	.429	.666
Fisher Ker Mag	.398	.387	.584
KL 20G	.386	.343	.495
GMM Posterior	.319	.376	.463

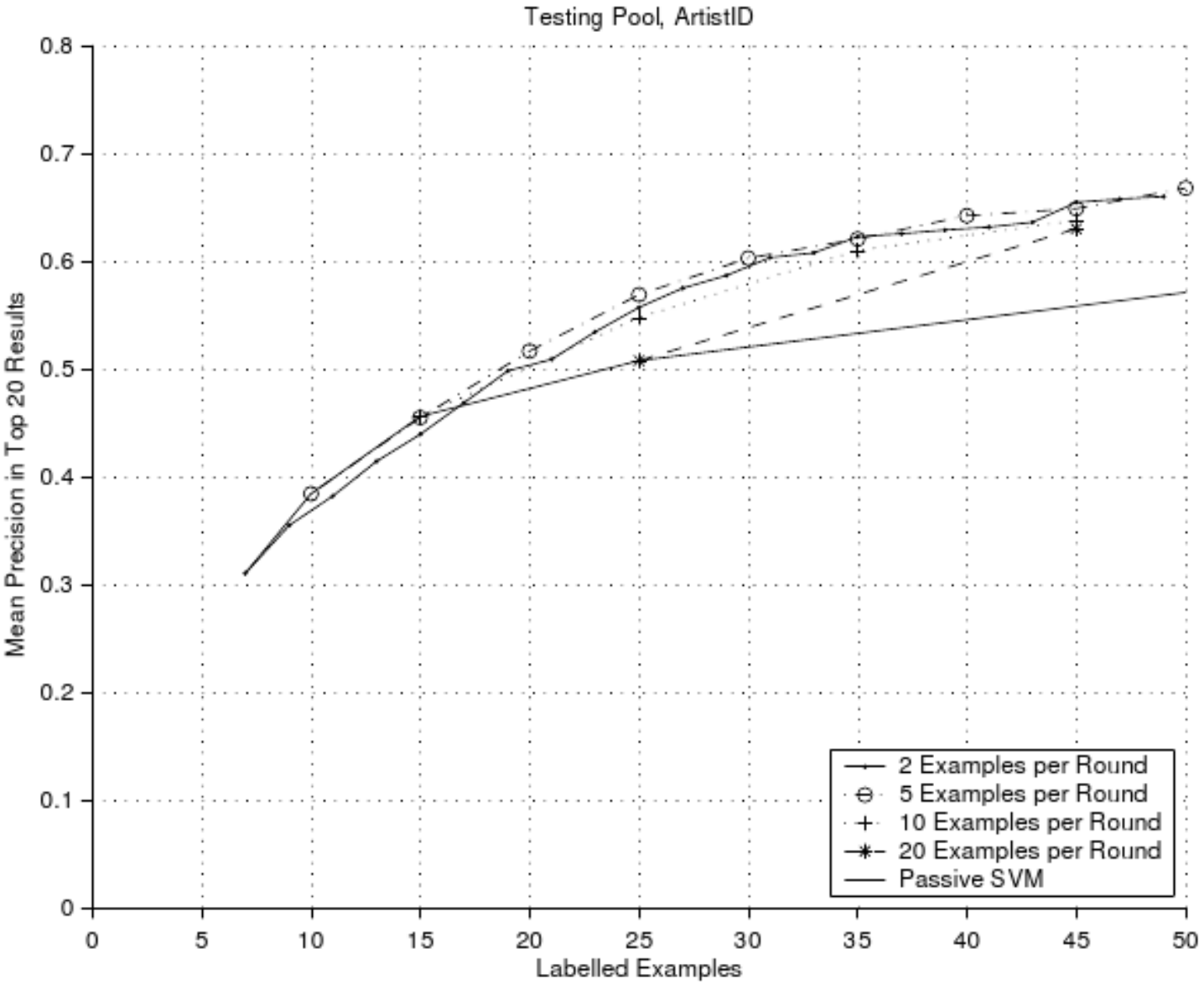
Experiment #2: Passive vs. Active Learning

- SVM trained with 50 examples in total
- One trial used passive learning (e.g. 50 randomly selected labeled songs)
- The remaining trials used active learning, and varied the number of examples shown per round:
    - 2 examples per round (25 rounds)
    - 5 examples per round (10 rounds)
    - 10 examples per round (5 rounds)
    - 20 examples per round (3 rounds)

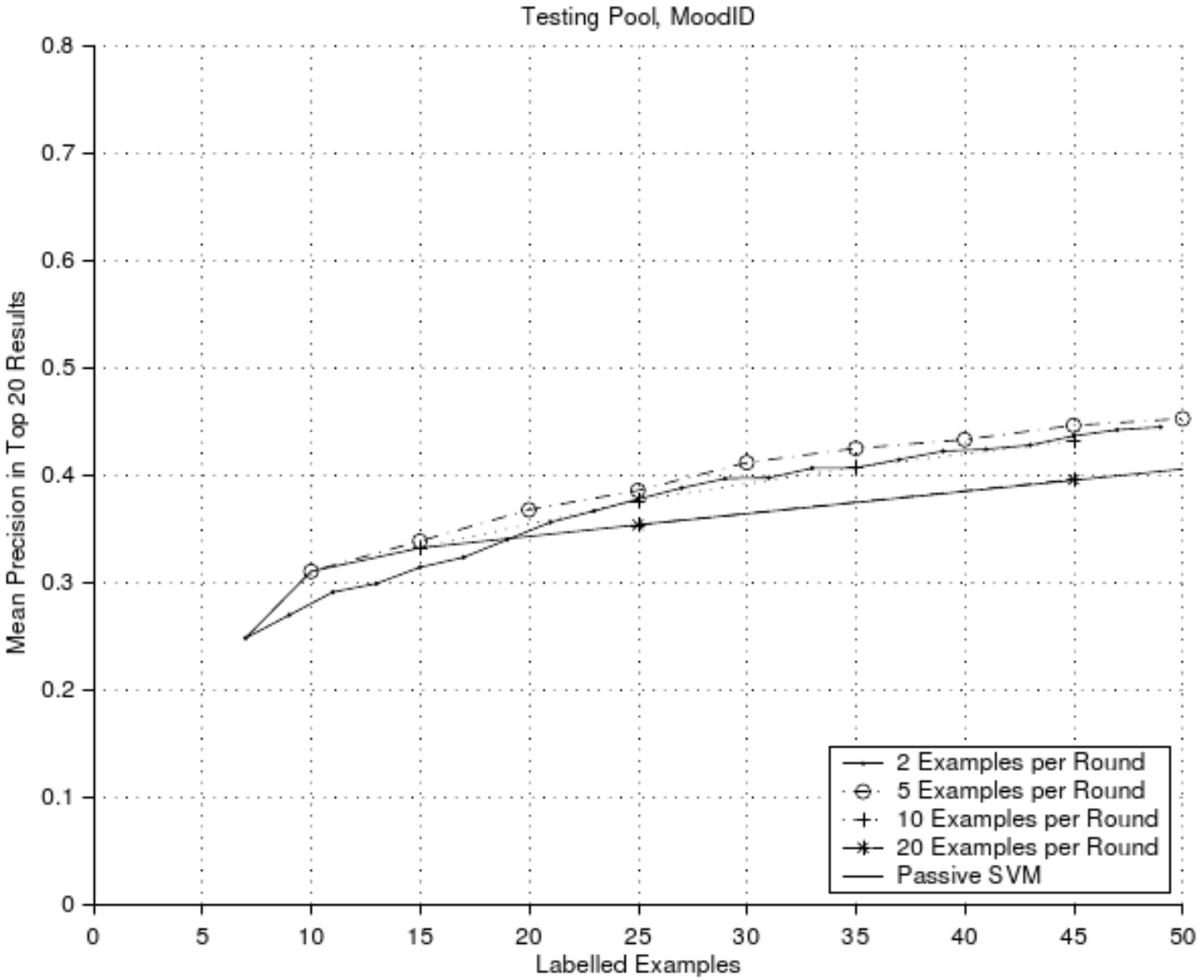
Experiment #2: Results

Ground Truth	Examples per round 2	5	10	20	Conv.
Style	.691	.677	.655	.642	.601
Artist	.659	.667	.637	.629	.571
Mood	.444	.452	.431	.395	.405

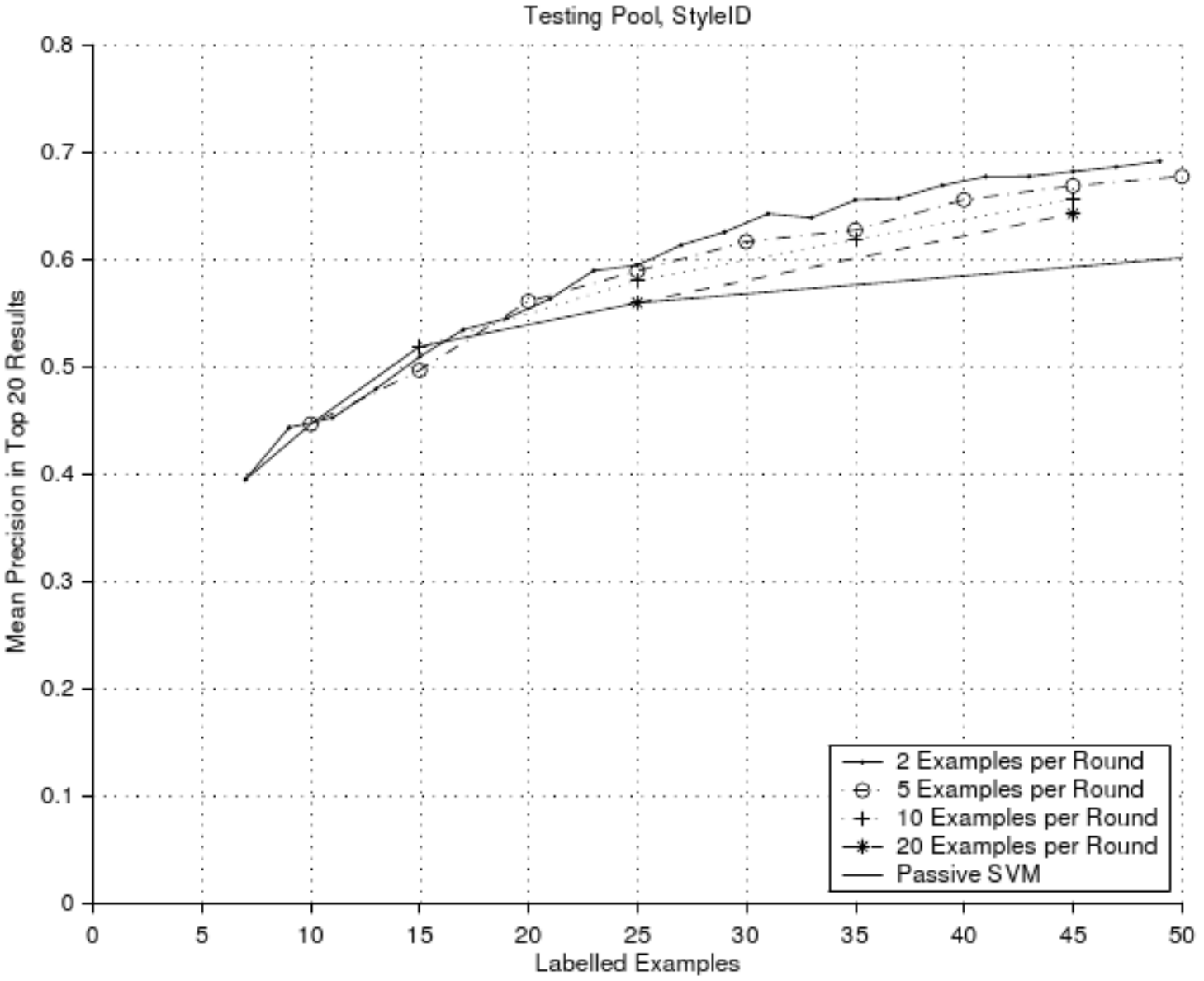
Experiment #2: Results



Experiment #2: Results



Experiment #2: Results



Future Work

- Use larger feedback sets for initial rounds?
- GMM features with KL divergence did poorly ... why?
- Performance degradation for GMM with 20 components did poorly ... why?
- Using relevance feedback in playlist generators (skipping a song interpreted as negative feedback)

Thank You