MSPR 7: Evaluation

Dr. Hendrik Purwins

AAU CPH

October 12, 2015

Outline

1. Introduction
2. Classification Evaluation
   - Cross-validation
3. Evaluation for Information Retrieval
   - Receiver Operating Characteristics
   - Examples/Exercises
   - Problem of True Labels
   - Summary Evaluation
4. Summary

Why classification?

- Processing chain: a) feature extraction, b) classifier, c) evaluation
- Image processing feature extraction:
  - Recognition of traffic signs through image processing in automatic robot car driving
  - Hand-writing recognition
  - Face recognition
- Audio processing feature extraction:
  - Automatic speech recognition
  - Speaker recognition
  - Automatic meeting transcription
  - Surveillance
  - Detection of harmony, meter, pitch in music
- Automatic fraud detection in online web shopping

Classification Evaluation

Why Evaluation?

Answer the questions:

- Does it really work?
- If it works somehow, how well does it work?
- How come it worked perfectly in the demo, and now I bought the tool and its complete crap?

Overfitting

- If the classification error is very small on the test data, but very large on the training data that could indicate overfitting.
- The principal goal of machine learning is to avoid overfitting, to build prediciton models that generalize to unknown data/situations.
- In overfitting, a model is fitted too perfectly to the training data (training data 'learned by heart') so it cannot generalize to new data (test data).
- Overfitting can occur if the model is too complex (e.g. to many parameters e.g. of covariance matrices to estimate) compared to very little, high-dimensional data.
- Overfitting can also occur if the training data is used for evaluation as well.

First Idea?

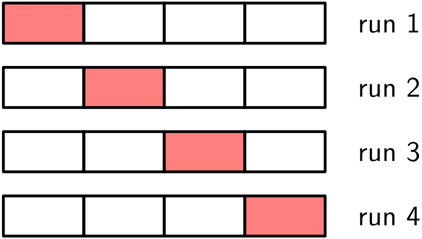
- Split our data into
  - *Training set*, from which to derive the parameters of the classifier (e.g. normal vector **w** of sparation hyperplane
  - *Test set*, on which the model is evaluated

- Works only once
- Wasteful of valuable training data!
- Enhanced strategy: *cross validation*.

Cross-validation

- Data is partitioned into S groups.
  - In each iteration, a different group is left out for validation.
  - The other $S-1$ groups are used for training.
- Example:

- Cross validation partitioning with $S=4$
- 4 runs possible.
- Red blocks: evaluation data.
- Efficient in data use!



PRTools Example

```
z=prdataset(X,species,...
 'featlab',['Petal Length'; 'Petal Width '],...
  'name','Fisher Iris'); % generate PRtools data
w = ldc([]); % initialize classifier with empty [] data
% 4-fold cross validation for data z, classfier object w
% returns error for all 4 cross validation runs
e = prcrossval(z,w,4);
a=1-e; % accurracy 0.96 (can vary due to randomization in cross val)
```

Which S to use?

- Dependence on
  - Data set size N, dimensionality of data, number of classes.
  - Calculation time (of single run and of grid search scheme)
- Extreme case $S = N$ (*leave-one-out cross validation*):
  - Maximizes the number of data points to built the model from
  - Maximizes computational cost (N training runs necessary on $N - 1$ data points)
  - Only one test data point (variance in the test results)
- Other Extreme: $S = 2$
  - Only half of the data used to build a model
  - Minimizing computational costs (2 training runs on $\frac{N}{2}$ data points)
  - Maximizing number of test data points per run(less variance in test results)

Other Aspects to Consider for CV

- Sometimes the single events in a block depend on each other, e.g. all the samples in one database or all the EEG recordings in one recording block.
- That is why it is advisable to take the entire block either for training or for testing. E.g. one entire sample database for training, the other one for testing, or leave-one-block out for EEG.

Example Musical Brain Computer Interface (Treder, Purwins, Miklody, Sturm, Blankertz 2014

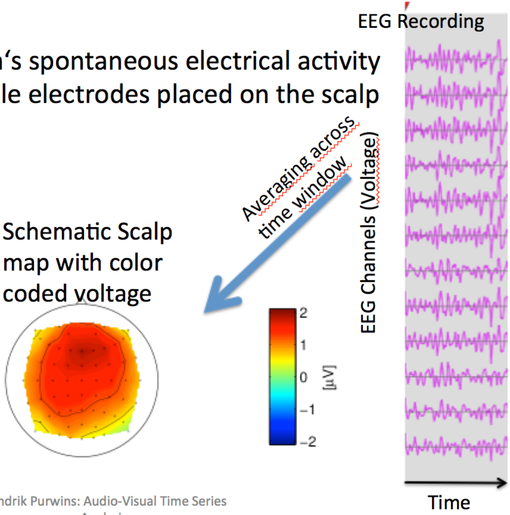
- A device (computer game, speller) is controlled by selectively switching the attention of the listener to one instrument or the other in a piece of simplified music
- Music listening allows for a potentially more pleasant and intuitive way of controlling a device

Electroencephalography (EEG)

- Recording of the brain‘s spontaneous electrical activity
- Recorded from multiple electrodes placed on the scalp

EEG cap

Schematic Scalp map with color coded voltage

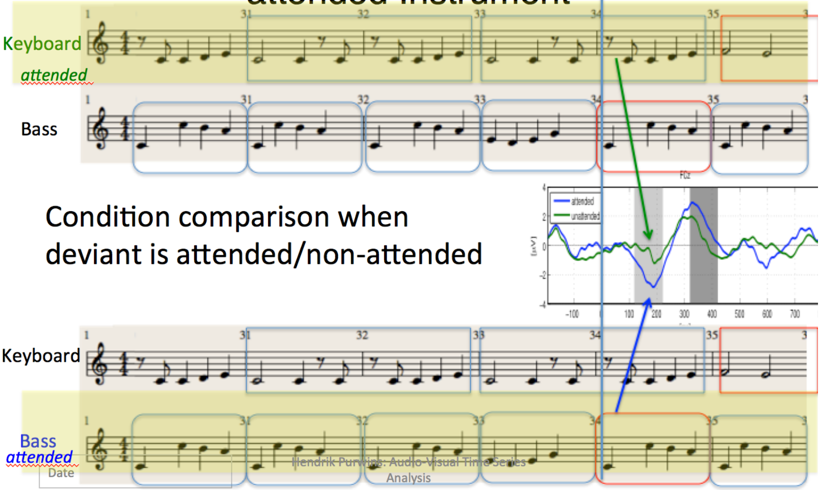


Date

Hendrik Purwins: Audio-Visual Time Series Analysis

Cross-validation

Classify 2 Conditions: Attended vs. Non-attended Instrument



Condition comparison when deviant is attended/non-attended

Date

Hendrik Purwins: Audio-Visual Time Series Analysis

Cross-validation

- Detect the changes of brain responses depending on the users attention using linear discriminant analysis.
- The stimuli blocks need to be shuffled to avoid bias through fatigue.

Other Means to Avoid Overfitting: Regularization

- Regularization (with regularization parameter λ adapts the complexity of the model (e.g. parameters in covariance matrix) to the size and dimensionality of the data. Regularization yields a compromise between quadratic classifier and minimum Mahalanobis classifier using the sample covariance matrices S_k for K classes:

$$\mathbf{S}_k^{reg} = (1-\lambda)\mathbf{S}_k + \frac{\lambda}{K}\sum_{k=1}^{K}\mathbf{S}_k$$

Class Assignment

How is the classifier called if $\lambda = 0$ and if $\lambda = 1$?

Shrinkage

Large eigenvalues are estimated too large and small eigenvalues are estimated too small. To counterbalance this bias a so called *shrinkage* of the covariance matrices towards the identity matrix I is introduced:

$$\hat{\Sigma}_i \mapsto (1-\gamma)\hat{\Sigma}_i + \gamma I \cdot \mathrm{trace}(\hat{\Sigma}_i)/n. \tag{1}$$

Regularization and shrinkage can also be combined which gives *regularized discriminant analysis (RDA)*

Confusion Matrix for Binary Classification

- True labels
  - Known through e.g. an expert or a measurement
- Predicted labels
  - Predicted by classifier

		Predicted Label	
		1	-1
True	1	True Positive (tp)	False Negative (fn)
Labels	-1	False Positive (fp)	True Negative (tn)

- False positive= Type I Error
  - 'Alternative hypothesis accepted, although not true'

Accuracy

		Predicted Label	
		1	-1
True	1	tp	fn
Labels	-1	fp	tn

- *Accuracy*:

$$\text{accuracy} = \frac{tp + tn}{tp + fp + fn + tn}$$

Example: Rate of correct classifications in a 2-class problem

Based on slides by Juan Jose Bosch

Recall, Precision, F-Measure

		Predicted Label	
		1	-1
True	1	tp	fn
Labels	-1	fp	tn

- *Recall (=sensitivity= true positive rate=d'=d-prime)* : proportion of correctly classified positives to actual positives

$$\text{recall} = \frac{tp}{tp + fn}$$

$$= P(\mathrm{pred.\ label} = 1 | \mathrm{true\ label} = 1)$$

Example: Rate of sick patients (true lab.=1) that are diagnosed as sick (pred. lab.=1)

- *Precision*: proportion of correctly classified positives to items predicted as positives

$$\text{precision} = \frac{tp}{tp + fp}$$

Example: Rate of patients, correctly diagnosed as sick to overall sick diagnoses.

- *F-measure*: compromise between precision and recall

$$F = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Recall, Precision, F-Measure

		Predicted Label	
		1	-1
True	1	tp	fn
Labels	-1	fp	tn

$$\text{recall} = \frac{tp}{tp + fn}$$

$$\text{precision} = \frac{tp}{tp + fp}$$

F-measure (=F-score=F_1-score): compromise between precision and recall

$$F = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Class Assignment

1. *Calculate the F-measure for a precision of 1 and a recall of 1*
2. *Calculate the F-measure for a recall of 0.5 and a precision of 0.5.*

Class Assignment

1. *Calculate the F-measure for a precision of 1 and a recall of 1*
2. *Calculate the F-measure for a recall of 0.5 and a precision of 0.5.*

$$\frac{2 \cdot 1 \cdot 1}{1+1} = \frac{2}{2} = 1$$

$$\frac{2 \cdot 0.5 \cdot 0.5}{0.5+0.5} = \frac{2 \cdot 0.25}{1} = 0.5$$

Maximum: 1(perfect classification), Minimum: 0 (worst classification)

Receiver Operating Characteristics (ROC) I

		Predicted Label 1	Predicted Label -1
True	1	True Positive (tp)	False Negative (fn)
Labels	-1	False Positive (fp)	True Negative (tn)

$$\text{recall} = \frac{tp}{tp + fn}$$

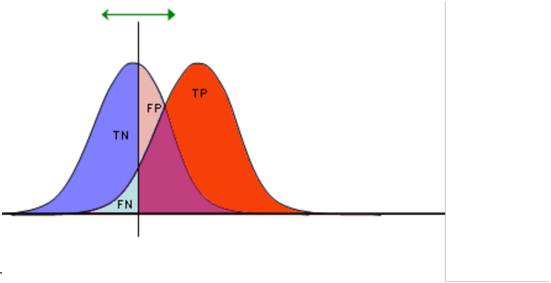
True label=1: something is wrong, True label=-1: everything is ok.
Pedicted label=1: Alarm goes off, predicted label=-1: nothing happens
False alarm rate = fall-out

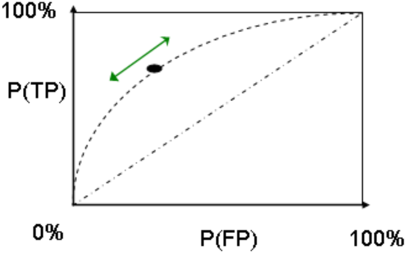
$$\text{far} = \frac{fp}{tn + fp} = P(\text{pred. label} = 1|\text{true label} = -1) \ \ \text{(false alarm rate)}$$

Based on slides by Juan Jose Bosch

Receiver Operating Characteristics (ROC) II

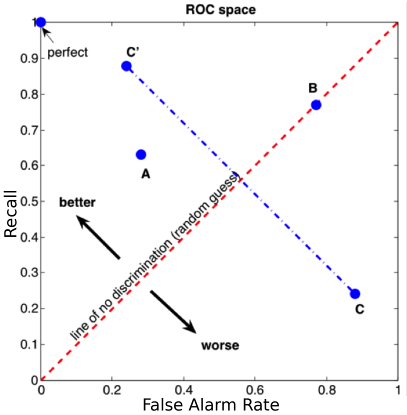
		Predicted Label	
		1	-1
True	1	tp	fn
Labels	-1	fp	tn





Receiver Operating Characteristics (ROC) III

Graphical plot recall $\frac{tp}{tp+fn}$ vs. false alarm rate far $= \frac{fp}{tn+fp}$.



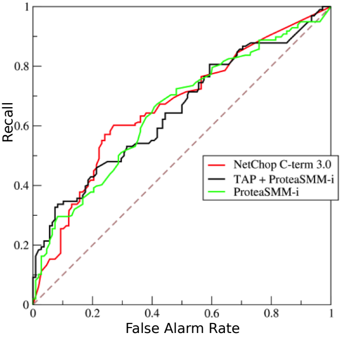
- Each classifier: 1 point in ROC space
- (0,0): classifier always predicts -1
- (1,1): classifier always predicts $+1$
- (0,1): perfect classification
- Worst results (1,0): inverted best results!

Receiver Operating Characteristics (ROC) III

- For a binary classifier, separation pane (e.g. in linear discriminant analysis) can be moved
  - From assigning class label '-1' to all items
  - To assigning class label '$+1$' to all items
  - and in between the two extremes.
- Procedure
  1. For different separation plane positions indicate the point (false alarm rate,recall) in the ROC space.
  2. Connect all these points to a line (ROC curve)
  3. Calculate *Area Under ROC Curve (AUC)*

Receiver Operating Characteristics

- Evaluating 3 HIV antigene predictors
- Recall vs. false alarm rate. separation border is is varied
- Area under the ROC curve (AUC) $\in [0;1]$
- Average performance across all possible discrimination borders
- Probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one



All three predictors better than random

Receiver Operating Characteristics

```
e = prroc(z,w); % test data z, trained classifier w
recall=1-e.xvalues
far= e.error
f=figure;
plot(far,recall)
err_auc=z*w*testauc
auc=1-err_auc;
```

Exercise

- Consider a classifiers δ_1
- Output of δ_1 is always $+1$ no matter what input.
- Apply the classifier to a dataset that consist of 80 items with label $+1$ and 20 items with label -1.
- Tasks:
  1. Write down the confusion matrix
  2. Calculate a) precision, b) recall, c) false alarm rate, d) accuracy, e) f-measure
  3. Comment on the results

Exercise Solution

- Data: 80 items with label $+1$ and 20 with label -1.
- δ_1 always gives $+1$.

1. Confusion Matrix:

		Predicted Labels	
		$+1$	-1
True	$+1$	80 (tp)	0 (fn)
Labels	-1	20 (fp)	0 (tn)

2. Calculate
   a. Precision $\frac{tp}{tp+fp} = \frac{80}{80+20} = 0.8$
   b. Recall $\frac{tp}{tp+fn} = \frac{80}{80} = 1$
   c. False positive rate $\frac{fp}{tn+fp} = \frac{20}{0+20} = 1$
   d. Accuracy $\frac{tp+tn}{tp+fn+fp+tn} = \frac{80+0}{80+20} = 0.8$
   e. F-Measure $\frac{2PR}{P+R} = \frac{2\cdot 0.80\cdot 1}{1+0.8} = \frac{1.6}{1.8} = \frac{8}{9}$

3. ROC curve: upper right corner (1,1)

5.
   - High values of evaluation measures
   - Due to unbalanced classes (80/20)
   - Performance comparable to random

Matlab PRTools Example Multi-Class Confusion Matrix I

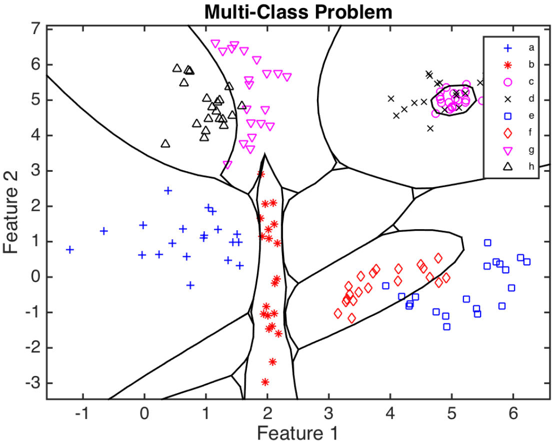
```
a = gendatm;  % load data
w = qdc(a);  %
d = a*w;     % classify training set
confmat(d); %display confusion matrix
cm=confmat(d);
acc=sum(diag(cm))/sum(sum(cm)) % calculate accuracy
```

PRTools Example Confusion Matrix II

```
 True   | Estimated Labels
 Labels |    a      b      c      d      e      f      g      h   |
--------|---------------------------------------------------------|
 a      |   20      2      0      0      0      0      0      0   |
 b      |    0     20      0      0      0      0      0      0   |
 c      |    0      0     18      2      0      0      0      0   |
 d      |    0      0      5     15      0      0      0      0   |
 e      |    0      0      0      0     18      2      0      0   |
 f      |    0      0      0      0      2     18      0      0   |
 g      |    0      0      0      0      0      0     20      0   |
 h      |    0      0      0      0      0      0      2     18   |
--------|---------------------------------------------------------|
 Totals |   20     20     23     17     20     20     22     18   |
```

Accuracy: 0.9187 Compare missclassifications to scatter plot on next page!

PRTools Example Confusion Matrix III



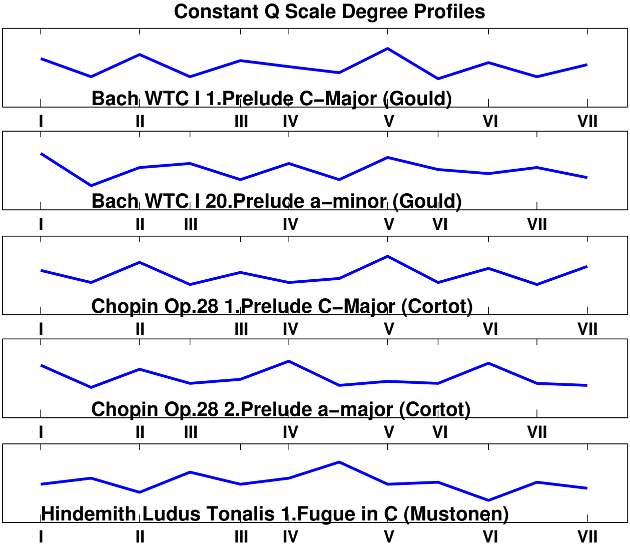
```
scatterd(a,'legend'); gridsize(30);  plotc(w);
```

Multiclass Evaluation

- Accuracy can be calculated for multiclass as well Just sum the diagonal elements of the confusion matrix and divide by the number of instances all together.
- F-measure and ROC /AUC are designed for two classes but they can be adapted for multi-class when one looks at each class individually class vs. all-other-classes. But this is not so straight forward.

Examples/Exercises

Classification of piano prelude composer based on the the accumulated energy of each pitch class, calculated from cq profiles (Purwins, Blankertz, Obermayer, 2001).



	LDA	RDA
Bach	0.79	**0.95**
Chopin	0.52	**0.64**
Alkan	0.43	**0.72**
Scriabin	0.65	**0.72**
Shostakovich	0.81	**0.86**
Hindemith	0.93	**0.97**

ROC AUC for 10-fold crossvalidation. Composer classification based piano preludes (represented as const Q profiles) one composer vs. the rest (Purwins, Blankertz, Obermayer 2004)..

Class Assignment

You want to build an onset detector to detect the onsets of the tones in a short musical excerpts. First you annotate the true onsets (in s) by listening carefully.

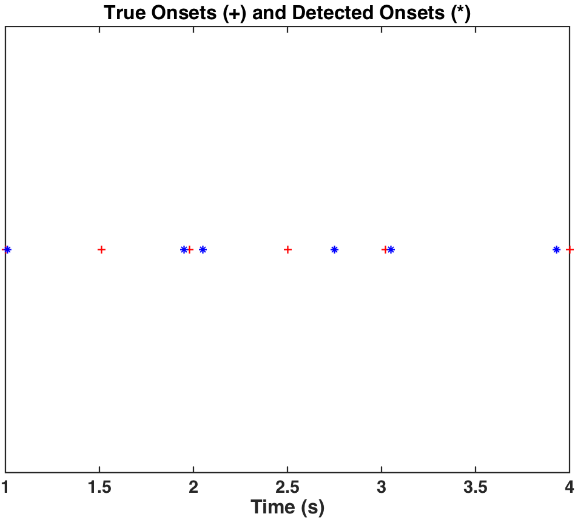
```
True_Onsets=[1.00 1.51 1.98 2.50 3.02 4.00];
```

Then you build an onset detection function that gives you the detected onsets below (in s)

```
Detected_Onsets=[1.01   1.95 2.05 2.75 3.05 3.93];
```

Think about how to evalute the preformance of the algorithm in terms of the f-measure. What problems need to be settled first before applying the f-measure?

Examples/Exercises



		Predicted Label 1	Predicted Label -1
True Labels	1	True Positive (tp)	False Negative (fn)
	-1	False Positive (fp)	True Negative (tn)

- Onset ist detected correctly if it is within a small time window (e.g. 50 ms) around a true onset.
- tp: number of correctly detected onsets
- fp: number of false positive onsets
- fn: number of missed onsets
- If there are 2 detected onsets within ±50ms of a true onset 1 wrong onset will be calculated.

Calculate the f-measure for the above numbers of detected and true onsets!

Examples/Exercises

```
   True_Onsets=[1.00  1.51  1.98  2.50  3.02  4.00];
   Detected_Onsets=[1.01   1.95  2.05  2.75  3.05  3.93];
```

- Tp: 1.01 1.95 3.05 (3)
- Double detection: 2.05
- Fp: 2.05 2.75 3.93 (3)
- Fn: (missed onsets): 1.51 2.50 4.00 (3)
- Precision: $\frac{Tp}{Tp+Fp} = \frac{3}{3+3} = \frac{1}{2}$
- Recall: $\frac{Tp}{Tp+Fn} = \frac{3}{3+3} = \frac{1}{2}$
- F-measure: $\frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision}+\text{Recall}} = \frac{2 \cdot \frac{1}{2} \cdot \frac{1}{2}}{\frac{1}{2}+\frac{1}{2}} = \frac{1}{2}$

Music Information Retrieval Evaluation eXchange (MIREX)

`http://www.music-ir.org/mirex/wiki/2015:Main_Page`

Categories e.g.:

1. Real-time Audio to Score Alignment (a.k.a Score Following)
2. Audio Chord Estimation
3. Onset Detection
4. Mood Classification
5. Audio Tempo Estimation

Class Assignment

How to evaluate a tempo detection algorithm?

MIREX II: Ground Truth Collection

1. Listeners were asked to tap to the beat of a series of musical excerpts. Responses were collected and their perceived tempo was calculated. (Moelants and McKinney, 2004).
2. For each excerpt, a distribution of perceived tempo was generated.
3. Simplification: The two highest peaks in the perceived tempo distribution for each excerpt were taken, along with their respective heights (normalized to sum to 1.0) as the two tempo candidates for that particular excerpt.

MIREX III: Example Tempo Estimation Data

1. http://www.music-ir.org/evaluation/MIREX/data/2006/tempo/
   User: tempo Password: t3mp0
2. For sound train2.wav, train2.txt contains slower tempo T1, faster tempo T2 and strength relation T1 relative to T2 e.g.

```
83.5   167.5  0.72
```

MIREX III:Evaluation

1. ST_1 : relative perceptual strength of slow tempo T_1 varies from 0 to 1.0
2. $TT_1 = 1$ if slower tempo has been identified within $\pm 8\%$, otherwise 0
3. $TT_2 = 1$ if faster tempo has been identified within $\pm 8\%$
4. Performance value: $P = ST_1 \cdot TT_1 + (1 - ST_1) \cdot TT_2$

MIREX IV:

Results

1. http://nema.lis.illinois.edu/nema_out/mirex2014/results/ate/summary.html
2. Reference paper of best algorithm by Sebastian Böck : http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.227.7572&rep=rep1&type=pdf

Example: Mood Classification - MIREX07 Dataset

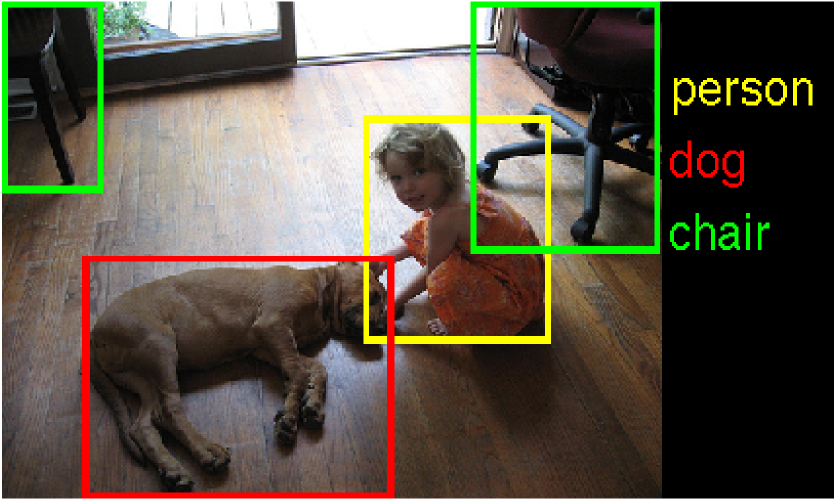
- Classify music audio according to the mood of the track
- 600 30 second audio clips in 22.05kHz mono WAV format
- 5 classes:
  1. passionate, rousing
  2. cheerful, fun, sweet
  3. bittersweet, autumnal
  4. humorous, silly
  5. aggressive, tense/anxious
- Runtime limitation
- Accuracy

IMAGENET Large Scale Visual Recognition Challenge 2015 (ILSVRC2015)

`http://image-net.org/challenges/LSVRC/2015` Categories:

1. Object detection
2. Object localization
3. Object detection from video
4. Scene classification

ILSVRC II: Object Detection Task



ILSVRC III: Data

1. Images hand labeled with presence /absence of 1000 object categories.
2. Training data: 1.2 million images
3. Validation and test data: 150000 photographs, collected from flickr etc.
4. 50 000: validation set

Problem of True Labels

- Sometimes it is expensive to get labels
- Labels can be wrong
  - Quantitiy of wrong labels determines quality of data set
- Labels can be ambiguous
  - e.g. 2 experts disagree on the correct label, e.g. a musical genre

Summary Classification Evaluation

- We want a classifier that generalizes to new data (no overfitting!)
- Never test on training data!
- Cross-validation is widely used, but still risk of overfitting.
- Various evaluation measures (confusion matrix, accuracy, F-measure)
- Receiver operating characteristics to average over shifted discrimination borders
- Costly labels

Lessons Learned Today

- How to evaluate classifiers (accurracy, f-measure, AUC-ROC, confusion matrix)
- How to make a classifier really learn something (e.g. by cross-validation to avoid overfitting)