Machine learning models for network related 3D video QoE

Lykourgiotis, A. Kordelas, I.Politis, C. Mysirlidis, T. Dagiuklas
University of Patras, Greece

(2nd ROMEO Workshop, Istanbul, 9/7/13)

SEVENTH FRAMEWORK PROGRAMME

Outline

- Aim of work in context of ROMEO
- QoE knowledge management
  - Machine learning categories
  - Naive Bayesian classifier
  - Logical decision tree
  - Multi-layer perceptron
- Performance evaluation
  - Simulation setup
  - Subjective evaluation
- Comparison of ML QoE models

Aim

- Goals of ROMEO project
  - delivery of live and collaborative 3D immersive media across next generation converged all-IP networks
  - development of a QoE based mobility management framework
    - handle horizontal and vertical handovers based on parameters and statistics from the application and underlying layers
- Provide a QoE-based mobility management across LTE-WiFi networks
  - MIH IEEE 802.21 framework
  - Handover decision
  - Stream adaptation

QoE Knowledge Management

- QoE estimation is an event based method
  - viewers respond and evaluate the perceptual (quality) experience by reflecting on the reactions that earlier events provoked
- Supervised ML
  - learning process based on instances produces a generalized hypothesis
  - forecasts future instances
- Main steps of ML techniques
  - gathering of the data set
  - data prepossessing
  - feature creation
  - algorithm selection
  - learning
  - test evaluation

Machine learning categories (1)

- Logic-based (decision trees)
  - nodes represent a feature of instances
  - branches represent a value that the node can assume
  - Disadvantage: not efficient if numerical features are used
- Perceptron-based (artificial neural networks)
  - It has been applied to a range of different real-world problems
  - Their accuracy is a function of the:
    - used number of neurons
    - processing cost
  - Disadvantage: inefficient when fed with irrelevant features

Machine learning categories (2)

- Statistical
  - Naive Bayesian classifier
    - requires short computational time for training
    - it distinguishes between classes using only a single Gaussian distribution
  - k-nearest neighbor
    - based on the fact that neighboring instances have similar properties
      - very simple to use it since it requires only the number of nearest neighbors
      - unreliable when applied on data sets with irrelevant features
- Support Vector Machines (SVM)
  - performs better when:
    - dealing with multi-dimension and continuous features
    - applied to inputs with a non-linear relationship between them

Naive Bayesian classifier (1)

- Specialized form of Bayesian network
  - Assumptions
    - predictive attributes are conditionally independent given the class
    - these are no hidden attributes that could affect the prediction
  - Properties
    - represents, uses and learns stochastic knowledge
    - accurately predicts the class of test instances given that the training instances include class information
- Statements
  - C - the class of an instance
  - c - a particular class label
  - X - the vector of a random variable that denotes the values of the attributes
  - x - a particular observed attribute value vector

Naive Bayesian classifier (2)

- Bayesian rule
  - computes the probability of each class given the vector of observed values for the predictive attributes

$$P(C=c|X=x) = \frac{P(C=c)P(X=x|C=c)}{P(X=x)} \quad (1)$$

- Naïve Bayesian
  - can be simple calculated by (2) since:
    - the event X = x
    - attributes are assumed to be conditionally independent

$$P(X=x|C=c) = \prod_i P(X_i = x_i|C=c) \quad (2)$$

- In case of continuous attributes
  - the probability density function for a normal (or Gaussian) distribution is

$$P(X=x|C=c) = G(x;\mu_c,\sigma_c), \quad (3)$$

$$G(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (4)$$

Logical decision tree (1)

- Decision trees can be:
  - a leaf node labeled with a class
  - a structure containing a test, linked to nodes
- Classification
  - instances are classified by applying the attribute vector
- C4.5 algorithm assumptions
  - when all cases belong to the same class
    - the tree is a leaf and is labeled with the particular class
  - calculate for every attribute the information gain that results from a test
    - according to the probability of each case having a particular value for the attribute
    - using the probabilities of each case with a particular value for the attribute being of a particular class
  - depending on the current selection criterion
    - find the best attribute to create a new branch.

Logical decision tree (2)

- C4.5 splitting criterion
  - normalized information gain
  - entropy $H(\vec{y})$ of the n-dimensional vector of attributes of the sample denotes the disorder on the data
  - conditional entropy $H(j \mid \vec{y})$ is derived from iterating over all possible values of $\vec{y}$
  - Goal:
    - find the attribute with the highest information gain and create a splitting decision node
    - prune the tree in order to minimize the classification error due to the outliers

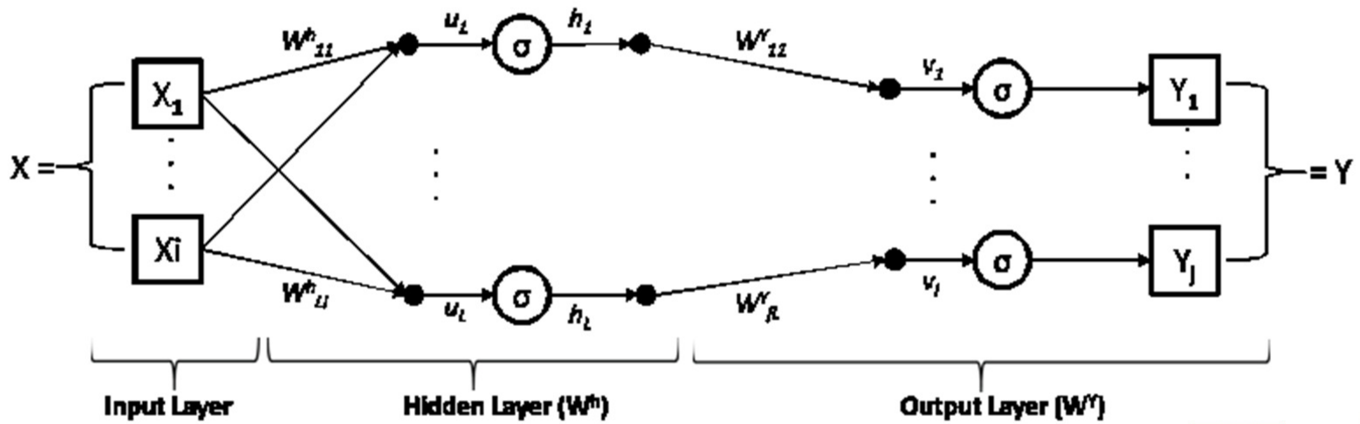
$$H(\overrightarrow{y}) = -\sum_{j=1}^{n} \frac{|y_j|}{|\overrightarrow{y}|} log \frac{|y_j|}{|\overrightarrow{y}|} \qquad (5)$$

$$H(j|\overrightarrow{y}) = \frac{|y_j|}{|\overrightarrow{y}|} log \frac{|y_j|}{|\overrightarrow{y}|} \qquad (6)$$

$$Gain(\overrightarrow{y}, j) = H(\overrightarrow{y} - H(j|\overrightarrow{y})) \qquad (7)$$

Multi-layer perceptron (1)

- Classification:
  i. Input layer of neurons distribute the values in the vector of predictor variable values, to the neurons of the hidden layers
  ii. hidden layers are fed with a bias of a constant input of 1.0
  iii. bias is multiplied by a weight and added to the sum going into the neuron
  iv. the weighted sum is fed to a transfer function
  v. the outputs from the transfer function are distributed to the output layer
  vi. the value from each hidden layer neuron is multiplied by a weight
  vii. the resulting weighted values are added together producing a weighted sum
  viii. the weighted sum is fed into the transfer function.
  ix. the output values of the transfer function are the outputs of the network



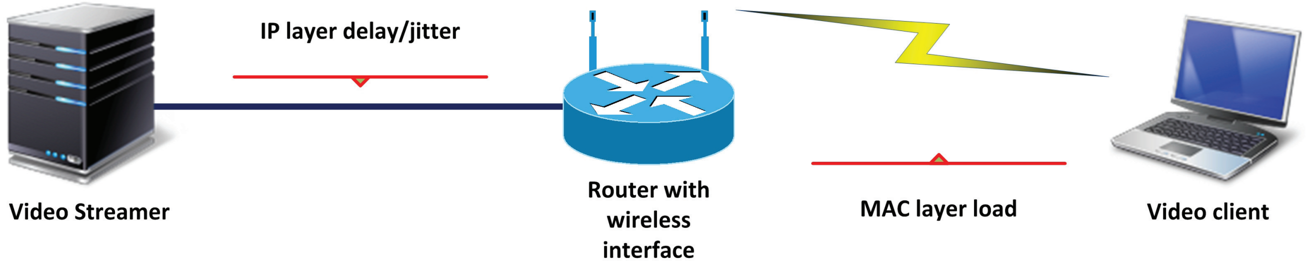
Multi-layer perceptron (2)

- Training process
  - determine the set of weight values that will result in a close match between the output from the neural network and the actual target values
- Algorithm precision depends on the number of neurons in the hidden layer
  - inadequate number of neurons
    - the network will be unable to model complex data and the resulting fit will be poor
  - too many neurons
    - the training time may become excessively long
    - the network may over fit the data
      - the network will begin to model random noise in the data
- Network parameters used:
  - six neurons
  - one hidden layer

Simulation setup (1)

- NS2
  - 802.11g WLANs extensions
- 3D video Sequences
  - Two left-right sequences, different spatial and temporal indexes
- Medium Grain Scale scalability
- RTP/UDP/IP protocol stack
  - MTU size of 1500 bytes

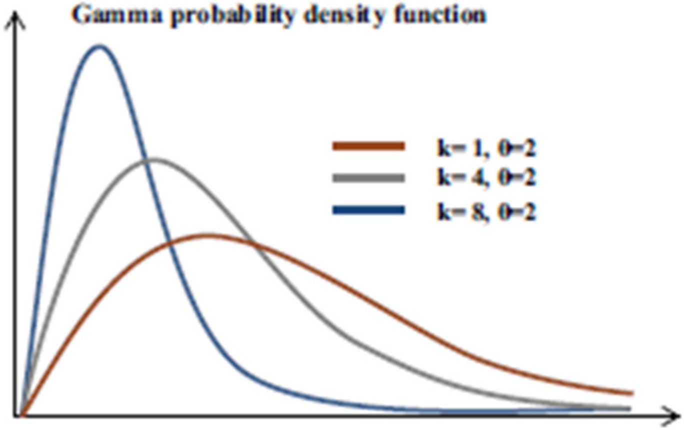
Video Sequence	Martial Arts	Munich
No of Frames	400	
Intra period	5 frames	
QP	(42,36) & (36,30)	
Frame rate	25 fps	
Resolution per view	640x720 pixels & 960x1080 pixels	
Spatial Index	21	25
Temporal Index	17	8



Simulation setup (2)

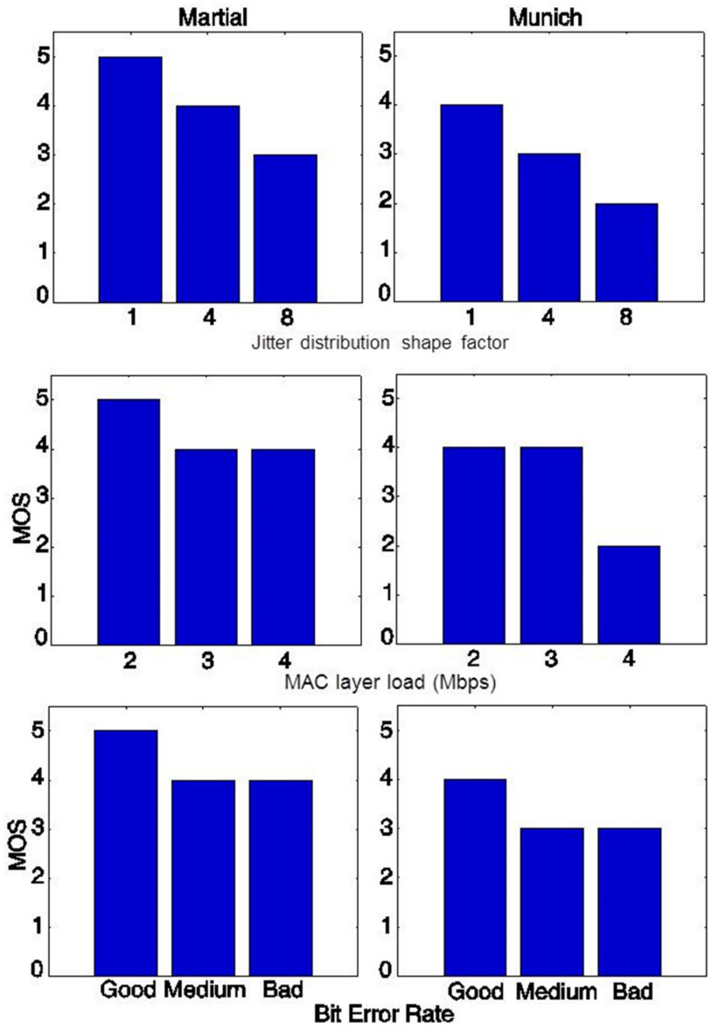
- Modeling the impact of wireless channel errors on the QoE
  - Rayleigh fading channel of the simulated 802.11g is represented by a two-state Markov model
- MAC layer load (time outs and retransmissions)
  - UDP traffic is transmitted to both uplink and downlink channels
  - Poisson distribution with a mean value of 2Mbps, 3Mbps and 4Mbps in each direction
- IP layer delay variation
  - constant plus gamma distribution
  - jitter increases based on the value of the shape parameter *k* and shape parameter *θ*

	Bad Channel	Medium Channel	Good Channel
P_G	$1.25e^{-2}$	$1.29e^{-2}$	$1.29e^{-2}$
P_B	$4.13e^{-14}$	$1.3e^{-13}$	$4.1e^{-12}$
P_{GG}	0.996	0.990	0.987
P_{BB}	0.336	0.690	0.740



Subjective evaluation

- Video sequences rating
  - Absolute category rating (ACR) method



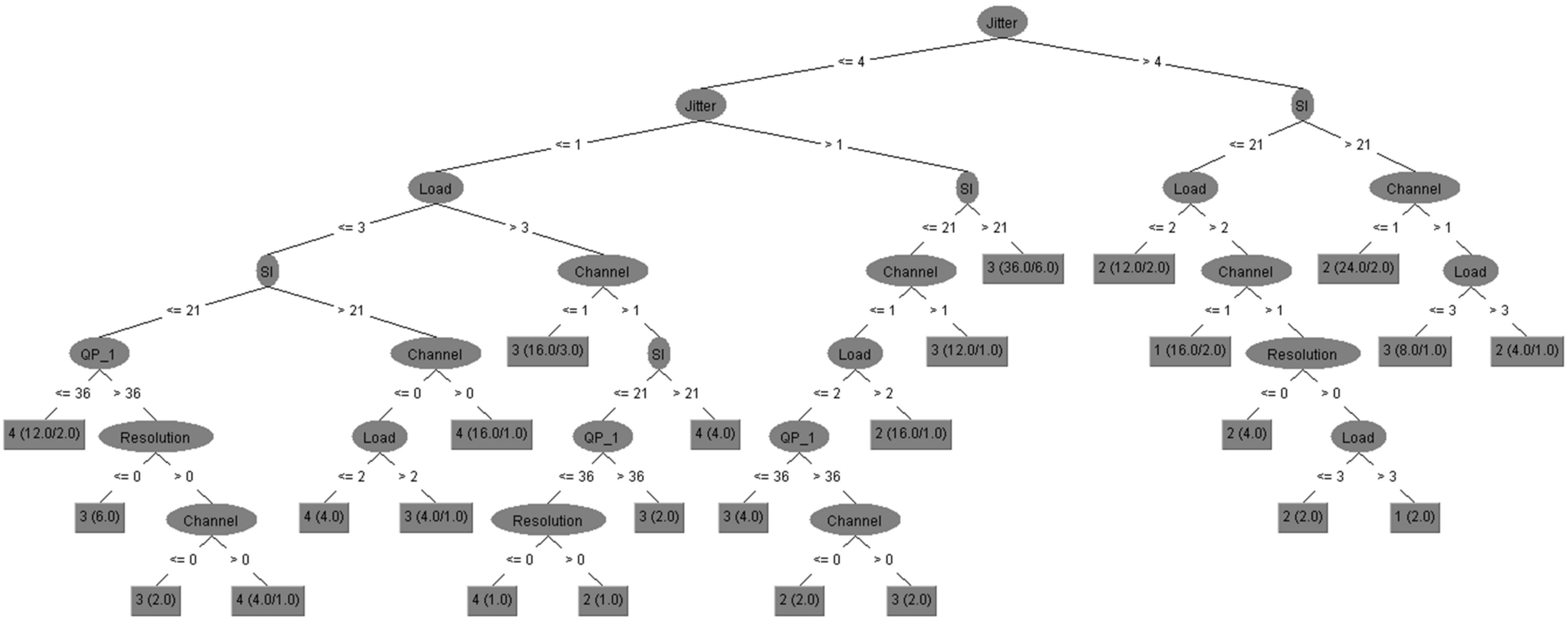
Naïve Bayes Classifier

- Output of the Naive Bayesian classification
  - implemented in Weka environment
  - mean and standard deviation of the Gaussian distribution for every attribute of the data set

Attributes		Class (MOS)				
		1	2	3	4	5
SI	mean	20.21	21.71	22.28	22.67	24
	std. dev.	0.89	1.97	1.98	1.88	0.6667
TI	mean	17.53	14.16	12.87	12	9
	std. dev.	2.01	4.45	4.45	4.24	1.5
QP_1	mean	39.15	38.82	39.55	38.14	36
	std. dev.	2.99	2.99	2.94	2.87	1
QP_2	mean	33.15	32.82	33.55	32.14	30
	std. dev.	2.99	2.99	2.94	2.87	1
Resolution	mean	0.57	0.51	0.46	0.5	1
	std. dev.	0.49	0.49	0.49	0.5	0.16
Jitter	mean	7	5.61	2.68	0.25	0
	std. dev.	0.58	2.00	2.17	0.90	0.5833
Load	mean	3.42	3.13	2.98	2.64	2
	std. dev.	0.67	0.82	0.81	0.71	0.16
Channel	mean	0.52	0.85	1.06	1.28	2
	std. dev.	0.67	0.77	0.83	0.76	0.16

Decision tree QoE prediction model

- Decision tree of the C4.5 ML
  - implemented as J48 in Weka environment
  - the jitter is the most important parameter

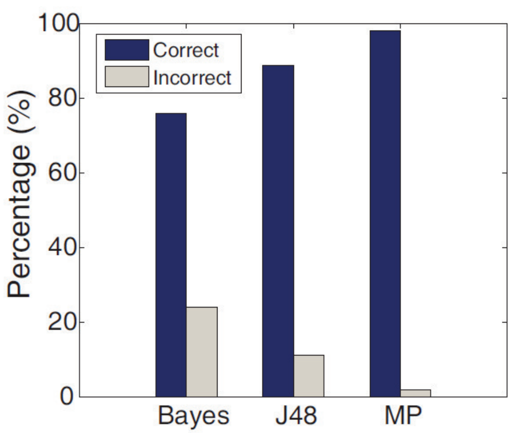


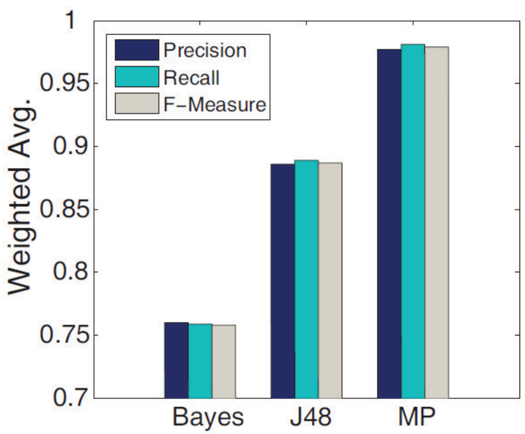
ROMEO

SEVENTH FRAMEWORK PROGRAMME

Comparison of ML QoE Models

- Algorithm's precision
  - denotes the degree to which repeated measurements under unchanged conditions show the same results
- Algorithm's recall
  - is defined as the number of relevant instances retrieved by a search divided by the total number of existing relevant instances
- Algorithm's F-measure
  - considers the precision and the recall





$$p = \frac{TP_i}{TP_i + FP_i}, r = \frac{TP_i}{TP_i + FN_i}, f = \frac{2 \cdot p \cdot r}{p + r}$$

I	Represents the class
TP_i	correctly classified instances
FP_i	instances that belong to the class i but they have not be classified there
FN_i	instances that do not belong to the class i but they have been classified there

Conclusions

- Currently considering three ML classification algorithms for modeling QoE due to network related impairments
- QoE model is a function of parameters collected not only from the application layer, but also from the underlying layers
- Packet loss as a function of a QoS parameters
- MOS comparison indicated that the predominant factor of QoE degradation is the IP layer delay variation
- Future work:
  - integrate the QoE classification model to the Handoff functionality and manage handover decision and mobility

Thank you!

Questions?