# Dynamic Mixture Ratio Model

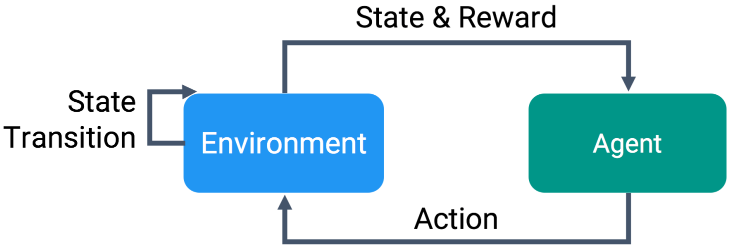
Marko Ruman, Miroslav Kárný

Institute of Information Theory and Automation,

Czech Academy of Sciences

Contacts: [ruman,school]@utia.cas.cz http://www.utia.cas.cz

# Decision making




The agents chooses actions in order to influence the environment and receive reward

# We wanted to create a very general model

To create a probabilistic model $\mathrm{P}(y|x,\Theta)$ that:

1. **generally approximates** any probability function / density,
2. can model any **mix of discrete and continuous** variables,
3. is capable of **online learning**,
4. is suitable for **decision making**.

$y$ - dependent variable, $x$ - explanatory variables, $\Theta$ - model parameters

# Markov property is assumed in decision making

## Data in decision making

$$D^{t-1} = (S_1, A_1, ..., S_{t-1}, A_{t-1})$$

Observed data contains **states** and **actions** for $t - 1$ epochs.

## Markov property

$$P(S_t | A_t, D^{t-1}, \Theta) = P(S_t | x_t, \Theta)$$

where $x_t$ is finite regression vector containing history of length $n$.
$x_t = A_t, S_{t-1}, A_{t-1}, ..., S_{t-n}, A_{t-n}$

# The inspiration comes from modelling of joint probabilitiy densities

## Gaussian mixture

A Gaussian mixture model is a universal approximator of smooth densities, [4].

## Modelling of mixed, **discrete-continuous** systems

$$X = (y, x) = (X_{D_1}, X_{D_2}, X_{C_1}, X_{C_2})$$

$$\begin{aligned} \mathrm{P}(X|\Theta) &= \mathrm{P}(X_{D_1}, X_{D_2}, X_{C_1}, X_{C_2}|\Theta) \\ &= \mathrm{P}(X_{C_1}, X_{C_2}|X_{D_1}, X_{D_2}, \Theta_1)\mathrm{P}(X_{D_1}, X_{D_2}|\Theta_2) \end{aligned}$$

The problematic modelling of a discrete variable depending on a continuous one is avoided.

# Ratio of Mixtures satisfies all of the goals

## The model is obtained from the joint probability

$$P(y|x,\Theta) = \frac{P(y,x|\Theta)}{\int_{\mathbf{y}} P(y,x|\Theta)}$$

## Modelling of joint $P(y,x|\Theta)$

$$P(y,x|\Theta) = \sum_{c \in \mathbf{C}} \alpha_c P_c(y, x_c|\Theta_c)$$

$\alpha_c$ - weight parameter of the $c$-th component

$P_c(y, x_c|\Theta_c)$ - the $c$-th mixture component

$x_c$ - the component specific regression vector

# Properties of Mixture ratios

## The resulting model has data-dependent weights

$$P(y|x,\Theta) = \sum_{c \in \mathbf{c}} w_c(x|\Theta) P_c(y|x_c,\Theta_c)$$

$w_c(x|\Theta)$ - weight of the $c$-th component depending on the data-vector $x$

Interesting properties:

- **Generally approximating**
- **Data-dependent weights**
- Modelling of mixed, **discrete-continuous** systems:
- Strong dependence on the **model structure**

# The mixture components are assumed to be from the exponential family

The specific choice of the mixture components $P_c$ is the following:

1. for $y$ and $x_c$ **discrete**
   $P_c(y, x_c|\Theta_c)$ is the categorical distribution
2. for $y$ **continuous**
   $P_c(y, x_c|\Theta_c)$ is the normal distribution
   - The component has separate parameters for each possible value of "discrete part" of $x_c$

# The online learning is done in two steps

The distribution on parameters $\mathrm{P}_t(\Theta)$ is evolved, $\Theta = (\alpha_c, \Theta_c)_{c \in \mathbf{c}}$.

$$1)\ \ \tilde{\mathrm{P}}_t(\Theta) \propto \mathrm{P}(y|x, \Theta)\mathrm{P}_{t-1}(\Theta) \qquad \text{(Bayes rule)},$$

$$2)\ \ \mathrm{P}_t = \operatorname*{argmin}_{\mathrm{P} \in \mathbf{P}} \mathrm{KL}(\tilde{\mathrm{P}}_t || \mathrm{P}) \qquad \text{(Kullback-Leibler projection)}$$

- The second step is to preserve the functional shape of $\mathrm{P}_t(\Theta)$.

$$\mathbf{P} = \left\{ \mathrm{P}(\Theta) \,\middle|\, \mathrm{P}_t(\Theta) = \mathrm{P}(\alpha) \prod_{c \in \mathbf{c}} \mathrm{P}(\Theta_c) \right\}$$

# The learning requires approximations

We need to approximate:

- The denominator of Bayes rule

$$\int_{\mathbf{\Theta}} \mathrm{P}(y|x,\Theta)\mathrm{P}_{t-1}(\Theta)d\Theta$$

- The integrations in Kullback-Leibler projection in the following form:

$$\int_{\mathbf{\Theta}} \tilde{\mathrm{P}}_t(\Theta)\ln\left(\mathrm{P}(\Theta)\right)d\Theta \qquad \mathrm{P}(\Theta) \in \mathbf{P}$$

## The learning requires approximations

The integrals are solved by
**Taylor expansion** around expected value of parameter $\hat{\Theta}$
It leads to solving a set of (numerically easily solvable) equations.

# A simulation suggests that Mixture Ratio outperforms the standard mixture model

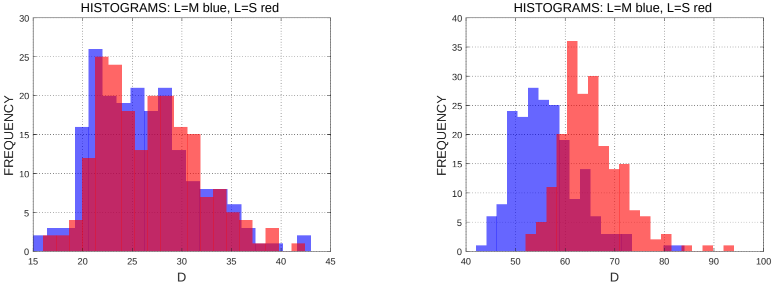



Figure: Histogram of $KL(P_R||P)$ with $P_R$ being the real probability of the system.
**Left**: system simulated by a standard mixture
**Right**: system simulated by mixture ratio with fixed parameters $\Theta$.

**Blue**: learning with mixture ratio — **Red**: learning with standard mixture

# Mixture Ratios show very good quality in real classification tasks

| | Logistic Regression | | Naive Bayes | | Mixture Ratio | |
|---|---|---|---|---|---|---|
| Dataset | Log loss | F1 | Log loss | F1 | Log loss | F1 |
| Iris | **0.18** | 0.93 | 0.53 | 0.54 | 0.22 | **0.95** |
| Wine | 0.51 | 0.85 | **0.50** | 0.80 | 0.69 | **0.89** |
| Cancer | **0.21** | 0.92 | 0.38 | 0.84 | 0.24 | **0.92** |
| UFC | 0.63 | 0.42 | **0.63** | 0.40 | 0.64 | **0.46** |
| Credit card fraud | **0.006** | 0.50 | 0.007 | 0.50 | 0.007 | **0.52** |

The values are obtained by 5-fold crossvalidation The datasets were discretized.

# A Python library is being developed

## Github repository

https://github.com/marko-ruman/mixture-ratio

## Code example

$variables\_domain = [10, 5, 7, 3, 2]$
$variables\_connection = [[0, 1, 2], [0, 3], [0, 4]]$

$mixture = MixtureRatio(variables\_domain, variables\_connection)$

$mixture.fit(X, Y)$

# Many things remain to be done

- Creating automatic optimal structure choice of the Mixture Ratio model
- Further developing the Python library - the continuous and mixed case is missing.
- Optimizing the speed of learning.

# Let's summarize the main points

- A **general probabilistic model** was created.
- A **Python library** is in development.

## Pros

- Generally approximating
- Online learning
- Ability to model discrete-continuous systems
- A lot of freedom in model structure

## Cons

- Computationally demanding learning
  - In every leaning step, a set of (convex) equations are solved numerically.

Thank you for your attention!


- M. Kárný, M. Ruman: *Mixture Ratio Modelling of Dynamic Environments*, Neural Networks, 2019, submitted.
- M. Ruman. *Mixture Ratios for Decision Making*. Master thesis, Czech Technical University, Prague, 2018.
- M. Kárný. *Approximate Bayesian recursive estimation*. Information Sciences, 289:100-111, 2014.
- G. McLachlan, D.Peel. *Finite Mixture Models*. Wiley Series in Probability and Statistics, 2000-11-01.


The research was supported by the research projects LTC18075 and CA16228