Active domain adaptation & astronomy

Ondřej Podsztavek

Faculty of Information Technology
Czech Technical University in Prague

Machine Learning and Modelling Seminar
March 16, 2023

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Motivation

Active domain adaptation

Predictive uncertainty

Future

Annotate a data set

- large *target* data set $T = \{\mathbf{x}_i\}_{i=1}^{n}$ with $n \in \mathbb{N}$ data samples
- annotations y_i
  - regression $y_i \subseteq \mathbb{R}$
  - classification $y_i \in \{1, \ldots, c\}$ where $c \in \mathbb{N}$

Redshift prediction

- T is Sloan Digital Sky Survey data release 16 quasar superset (hereafter DR16Q superset)
- composed of spectra $\mathbf{x}_i \in \mathbb{R}^{3752}$
- $n = 1440573$
- regression problem, i.e. $y_i \in \mathbb{R}$ is redshift

What is Sloan Digital Sky Survey (SDSS)?

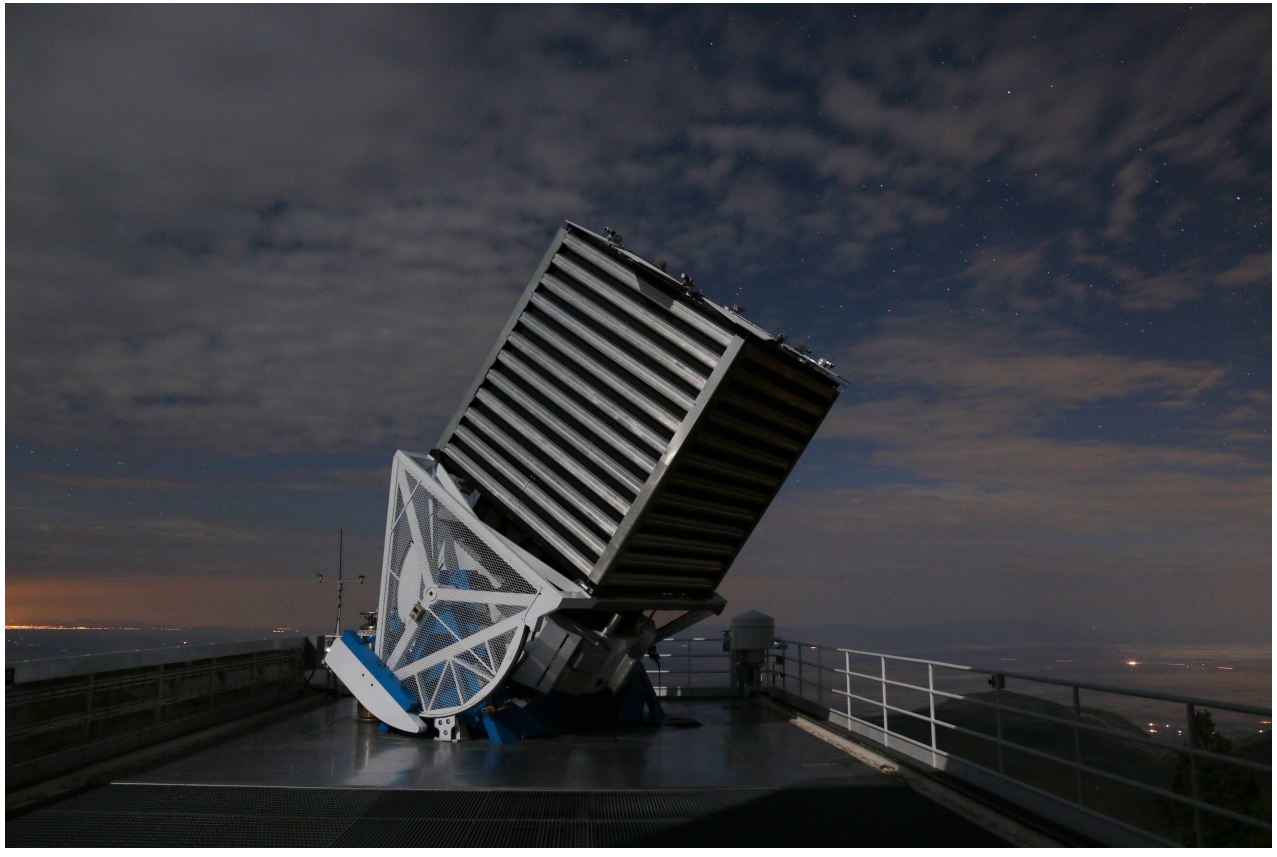
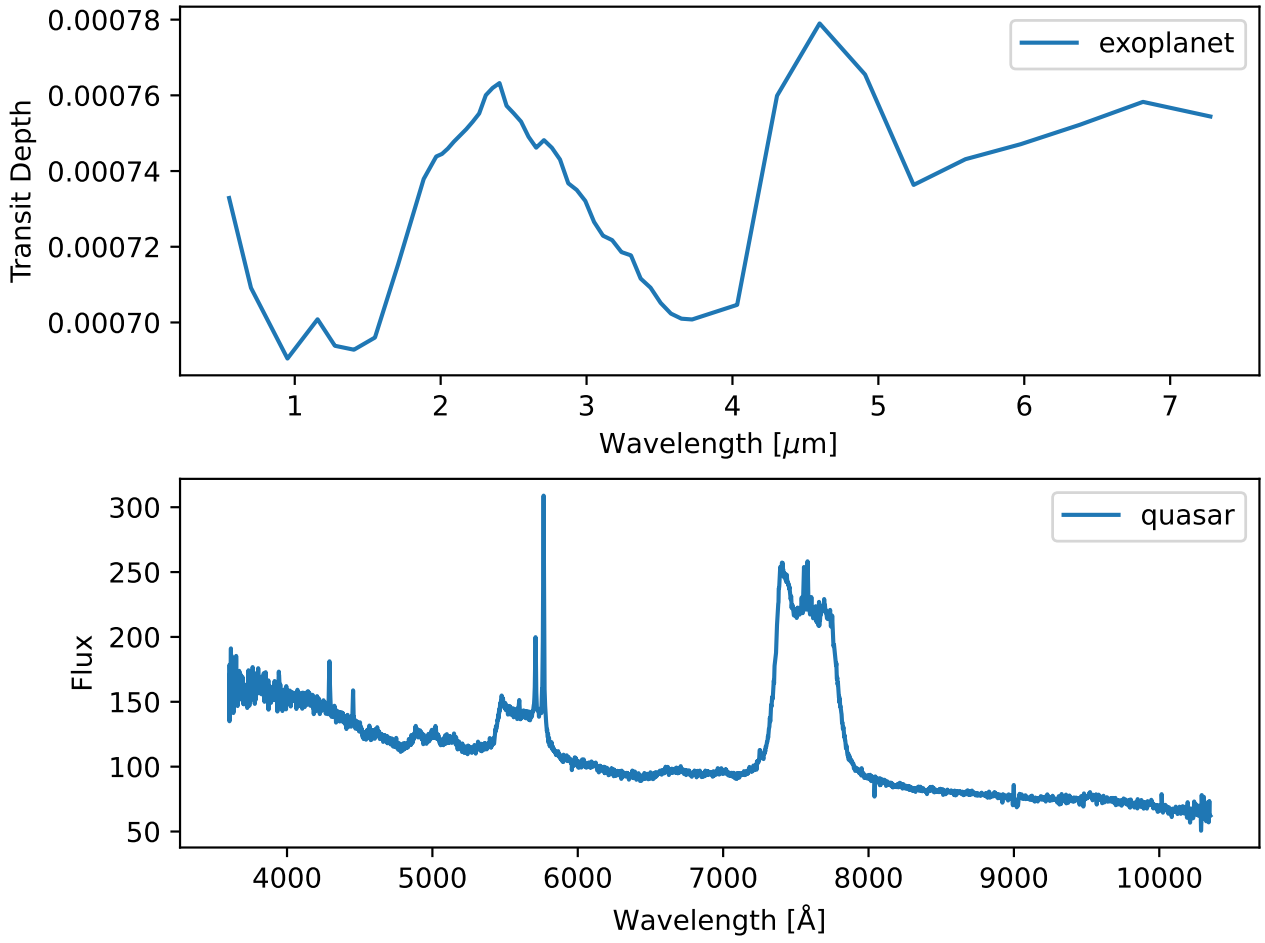


Figure: The SDSS telescope located in the Apache Point Observatory, New Mexico. Photo by Patrick Gaulme is licensed under CC BY 4.0.

What is spectrum?



What is quasar (QSO)?

- QSOs are the most luminous objects in the universe
- due to their enormous distances, they appear like stars
- so they are called quasi-stellar objects
- wider class of *active galactic nuclei* where the energy is produced by the accretion onto the supermassive black hole

What is spectroscopic redshift?

- measuring redshift of an object from its spectrum by locating atomic emission and absorption lines
- ratio of wavelength offset to the original laboratory wavelength is called a (cosmological) redshift

Why is redshift prediction important?

redshifts of galaxies and QSOs are used in cosmology

- expansion of the universe
- question whether space is finite or infinite

Human annotations

- *large* data set T so we cannot get human annotations
- slow (infeasible) but very reliable
- we want humans to focus on more important tasks

Redshift prediction

- human annotated up to the DR12Q superset
- but DR16Q superset has $n = 1440573$

Automated annotations

- we must rely on automated methods
- automated methods might produce incorrect annotations
- i.e. unreliable in some (e.g. extreme) cases.

Redshift prediction

- automated methods, e.g. template fitting
- redhifts by template fitting $y_i > 5$ should be considered suspect in the DR16Q superset

How to identify incorrect annotations?

- predictive uncertainty associated with annotations
- consistency check, i.e. diverse set of automatic methods

then

1. check data with inconsistent values ordered by their predictive uncertainties
2. check consistent values with higher predictive uncertainties.

Machine learning

machine learning method (especially deep learning) as an automated method

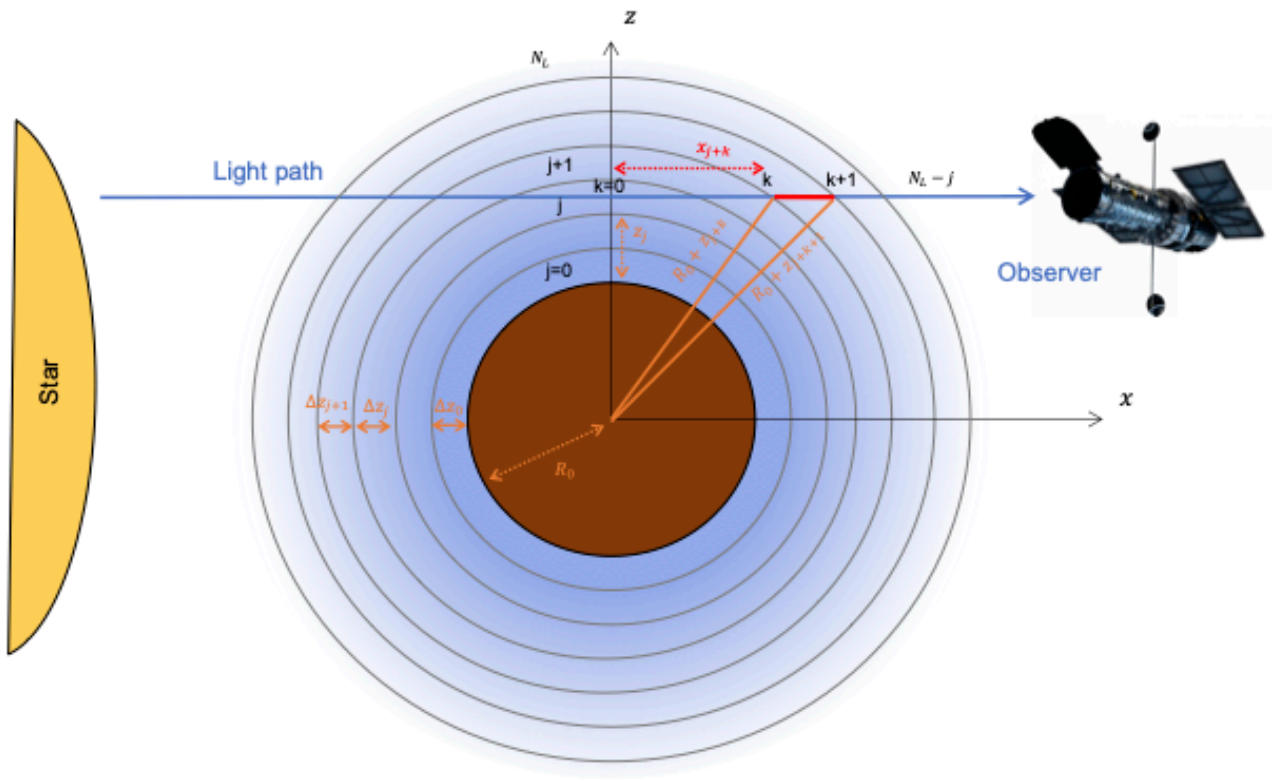
- might mimic (substitute) a human annotator
- if trained with human annotated traning set $S = \{(\mathbf{x}_i, y_i)\}_{i=n+1}^{n+m}$ with $m \in \mathbb{N}$ pairs of data samples

Redshift prediction

- Bayesian convolutional neural network
- there is the older DR12Q superset with 523331 human annotated spectra

Machine learning is fast

- Ariel Data Challenge to characterise exoplanets' atmospheres
- Markov Chain Monte Carlo methods and nested sampling method are slow



Machine learning problem

target data set T and older training data set S have different distributions

Redshift prediction

DR12Q superset is not i.i.d. sample of the DR16Q superset

Formalisation: Target set

- large unannotated target set $T = \{\mathbf{x}_i\}_{i=1}^{n}$
- $n \in \mathbb{N}$ data samples $\mathbf{x}_i \in \mathcal{X} = \mathbb{R}^d$ where $d \in \mathbb{N}$
- T is a realization of random sample of n i.i.d. random vectors $\mathbf{X}_T$ with joint probability density function (PDF) $f_{\mathbf{X}_T}(\mathbf{x})$.
- annotations y_i
  - $y_i \in \mathcal{Y} \subseteq \mathbb{R}$ in regression
  - $y_i \in \mathcal{Y} = \{1, \ldots, c\}$ where $c \in \mathbb{N}$ in classification

Formalisation: Source set

- human annotated source set $S = \{(\mathbf{x}_i, y_i)\}_{i=n+1}^{n+m}$
- $m \in \mathbb{N}$ pairs of data instances $\mathbf{x}_i$ and human annotations y_i
- $\{\mathbf{x}_i\}_{i=n+1}^{n+m}$ of S are realization of random sample of m i.i.d. random vectors $\mathbf{X}_S$ with joint PDF $f_{\mathbf{X}_S}(\mathbf{x})$
- $f_{\mathbf{X}_S}(\mathbf{x}) \neq f_{\mathbf{X}_T}(\mathbf{x})$

Supervised machine learning

- learn a conditional PDF $f(y|\mathbf{x}, \boldsymbol{\theta})$
- approximate the true conditional PDF $f_{Y_T|\mathbf{X}_T}(y|\mathbf{x})$
- suppose $f_{Y_S|\mathbf{X}_S}(y|\mathbf{x}) = f_{Y_T|\mathbf{X}_T}(y|\mathbf{x})$
- implies $f_{\mathbf{X}_S, Y_S}(\mathbf{x}, y) \neq f_{\mathbf{X}_T, Y_T}(\mathbf{x}, y)$

Target generalisation error

we want to minimize the target generalisation error

$$L^*_{(\mathbf{X}_T, Y_T)}(\boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{x},y)\sim(\mathbf{X}_T, Y_T)} l(f(y|\mathbf{x}, \boldsymbol{\theta}), y)$$

where l is a *proper scoring rule*

Source empirical error

we can train a model with

$$L_S(\boldsymbol{\theta}) = \frac{1}{m} \sum_{i=n+1}^{n+m} l(f(y|\mathbf{x}_i, \boldsymbol{\theta}), y_i) \approx L^*_{(\mathbf{X}_S, Y_S)}(\boldsymbol{\theta})$$

that approximates the source generalisation error

Transfer learning

given

- source domain $(\mathcal{X}_S, f_{\mathbf{X}_S}(\mathbf{x}))$ and task $(\mathcal{Y}_S, f_{Y_S|\mathbf{X}_S}(y|\mathbf{x}))$
- target domain $(\mathcal{X}_T, f_{\mathbf{X}_T}(\mathbf{x}))$ and task $(\mathcal{Y}_T, f_{Y_T|\mathbf{X}_T}(y|\mathbf{x}))$

transfer learning aims to improve the learning of the target task in the target domain using knowledge from the source domain and the source task.

Transfer learning in context of CNNs

initialisation with pretrained parameters on S followed by fine-tuning with target set (supposes annotations in target set)

Domain adaptation and covariate shift

in domain adaptation domains are different

- $\mathcal{X}_S = \mathcal{X}_T$ (homogenous) or $\mathcal{X}_S \neq \mathcal{X}_T$ (heterogenous)
- covariate shift $f_{\mathbf{X}_S}(\mathbf{x}) \neq f_{\mathbf{X}_S}(\mathbf{x})$
- $\mathcal{Y}_S = \mathcal{Y}_T$
- $f_{Y_S|\mathbf{X}_S}(y|\mathbf{x}) = f_{Y_T|\mathbf{X}_T}(y|\mathbf{x})$

Domain shift

- caused by change in the measurement system
- annotation depends of latent variable
- we observe some representation of it

Categorisation of domain adaptation

Unsupervised domain adaptation

given a annotated source set S and an unannotated target set T learn $f(y|\mathbf{x}, \theta)$.

Instance-based domain adaptation

reduces distributions difference by reweighting the source samples and it trains on the weighted source samples

Feature-based domain adaptation

a common shared space is generally learned in which the distributions of the two datasets are matched

Data importance-weighting

we can do importance-weighting to train model

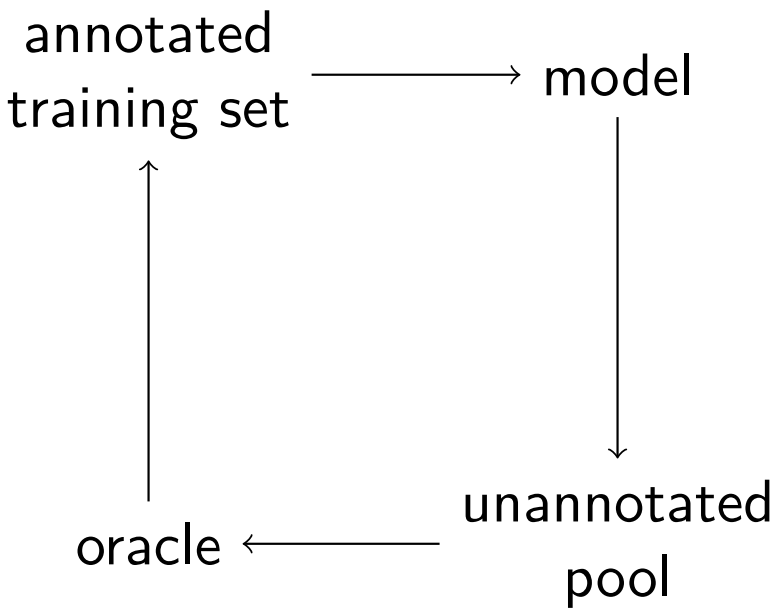
$$
\begin{aligned}
L^*_{(\mathbf{X}_T, Y_T)}(\boldsymbol{\theta}) &= \mathbb{E}_{(\mathbf{x},y)\sim(\mathbf{X}_T, Y_T)} l(f(y|\mathbf{x}, \boldsymbol{\theta}), y) \\
&= \mathbb{E}_{(\mathbf{x},y)\sim(\mathbf{X}_T, Y_T)} \frac{f_{\mathbf{X}_S, Y_S}(\mathbf{x}, y)}{f_{\mathbf{X}_S, Y_S}(\mathbf{x}, y)} l(f(y|\mathbf{x}, \boldsymbol{\theta}), y) \\
&= \mathbb{E}_{(\mathbf{x},y)\sim(\mathbf{X}_S, Y_S)} \frac{f_{\mathbf{X}_T, Y_T}(\mathbf{x}, y)}{f_{\mathbf{X}_S, Y_S}(\mathbf{x}, y)} l(f(y|\mathbf{x}, \boldsymbol{\theta}), y) \\
&= \mathbb{E}_{(\mathbf{x},y)\sim(\mathbf{X}_S, Y_S)} \frac{f_{\mathbf{X}_T}(\mathbf{x})}{f_{\mathbf{X}_S}(\mathbf{x})} l(f(y|\mathbf{x}, \boldsymbol{\theta}), y) \\
&\approx \frac{1}{m} \sum_{i=n+1}^{n+m} w(\mathbf{x}_i, \boldsymbol{\iota}) l(f(y|\mathbf{x}_i, \boldsymbol{\theta}), y_i)
\end{aligned}
$$

Data importance-weighting problem

- division by zero if $(\mathbf{x}, y) \sim (\mathbf{X}_T, Y_T)$ such that $f_{\mathbf{X}_S, Y_S}(\mathbf{x}, y) = 0$
- model will not be trained on $\mathbf{x}_i \in T$ but $\mathbf{x}_i \notin S$
- how much we can generalise?

Active learning

- model will perform better if it can to choose data for training
- model queries unannotated data samples to be annotated by an oracle



Active domain adaptation

- select samples that help overcome the distribution mismatch between a source and target distributions
- selecting an informative subsample for annotation is crucial
  1. contains data with high model uncertainty
  2. is diverse, but
  3. not redundant

Discovery of objects of interest

Škoda P., Podsztavek O., Tvrdík P., 2020. Active deep learning method for the discovery of objects of interest in large spectroscopic surveys. *Astronomy & Astrophysics* 643, A122.

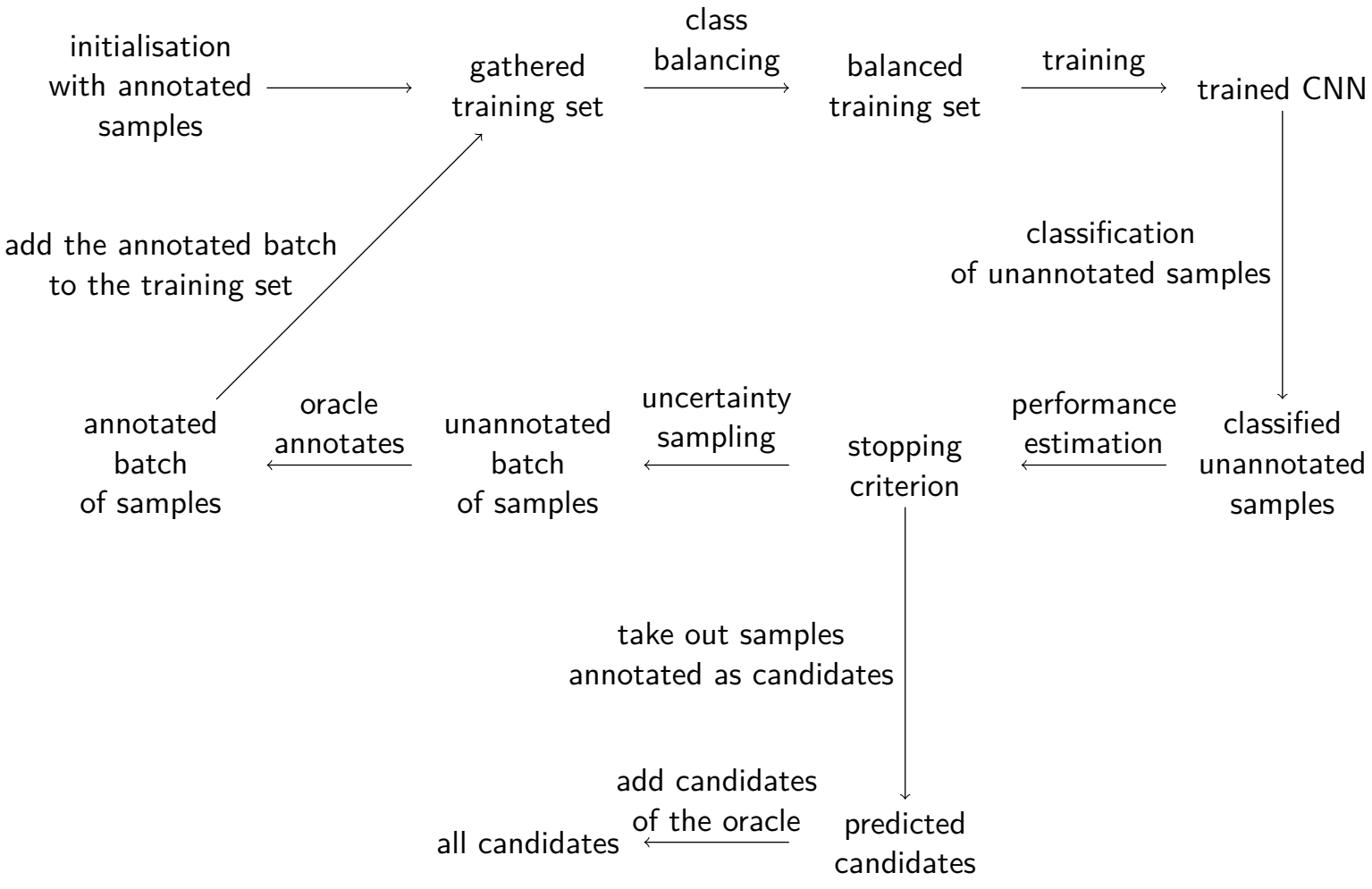
Motivation

discover *rare* (e.g. hard to model complicated physics) objects of interest in large spectroscopic surveys

Classical methods and their problems

- matching of template spectra
  → templates for rare objects of interest are not known
  → fails for complex shapes of only several spectral lines
- computes integral statistics
  → poor results with a high rate of false candidates

Flowchart of the active deep learning method



Experimental data of the active deep learning method

- goal is to identify emission-line spectra in 4.1 million spectra from the LAMOST survey
- initial training set is 13000 spectra from the Ondřejov telescope
- difference between Ondřejov and LAMOST telescope was compensated by Gaussian blurring

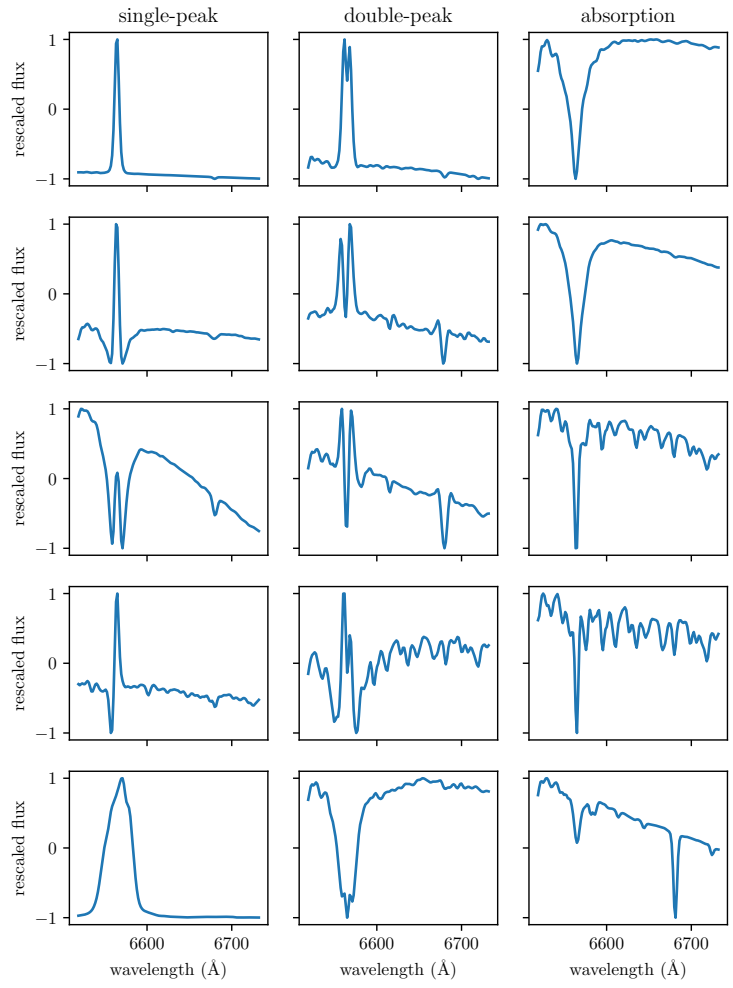


Figure: example Ondřejov spectra

Results of the active deep learning method

- the method identified 4379 emission-line spectra in the 4.1 million LAMOST spectra
- among them, 948 are *unknown rare* objects

Table: partial confusion matrix

predicted class	visually confirmed class		
	single-peak	double-peak	uninteresting
single-peak	3484	53	37
double-peak	18	548	21

Predictive uncertainty

- model confidence in predictions
- model outputs conditional PDFs not only point estimates
- predictive uncertainty can be decomposed into
  1. data uncertainty
  2. model uncertainty
  3. distributional uncertainty

Data (aleatoric) uncertainty

- noise inherent in the observations (homoscedastic and heteroscedastic)
  - noisy data samples $\mathbf{x}_i$
  - annotations y_i (e.g. class overlap or measurement error)
- irreducible even if we enlarge the training set with more data

Model (epistemic) uncertainty

- uncertainty if model parameters were learnt correctly
- modelled as probability distributions over model parameters
- but there also is uncertainty about model structure and training procedure
- can be reduced with more training data

Distributional uncertainty

- due to mismatch between the training (i.e. source) and target distributions
- can be modeled through model uncertainty

Approximations to Bayesian neural networks

Monte Carlo (MC) dropout

training of neural networks with dropout is equivalent to Bayesian variational inference

Deep ensembles

train an ensemble of models that output normal distributions

Calibration and sharpness

Gneiting et al. (2007): "maximizing the sharpness of the predictive distributions subject to calibration"

Calibration

statistical consistency between a conditional distribution and its annotation

Sharpness

concentration of the conditional distribution

Evaluation: Proper scoring rules

- scoring rules s assign a numerical score to a pair of distributions P_1 and P_2 (might be a degenerate distribution)
- proper scoring rule if $s(P_1, P_1) \geq s(P_1, P_2)$
- if the maximum is unique then it is *strictly* proper scoring rule

Evaluation: Negative log likelihood (NLL)

NLL is a proper scoring rule

$$\text{NLL}(f(y|\mathbf{x}_i, \boldsymbol{\theta}), y_i) = -\log f(y_i|\mathbf{x}_i, \boldsymbol{\theta})$$

Evaluation: Continuous ranked probability score (CRPS)

distance between conditional cumulative distribution function (CDF) $F(y|\mathbf{x}_i, \boldsymbol{\theta})$ and target value y_i:

$$\text{CRPS}(F(y|\mathbf{x}_i, \boldsymbol{\theta}), y_i) = \int_{-\infty}^{\infty} (F(y|\mathbf{x}_i, \boldsymbol{\theta}) - H(y - y_i))^2 dy$$

where $H(y)$ is the Heaviside step function defined as:

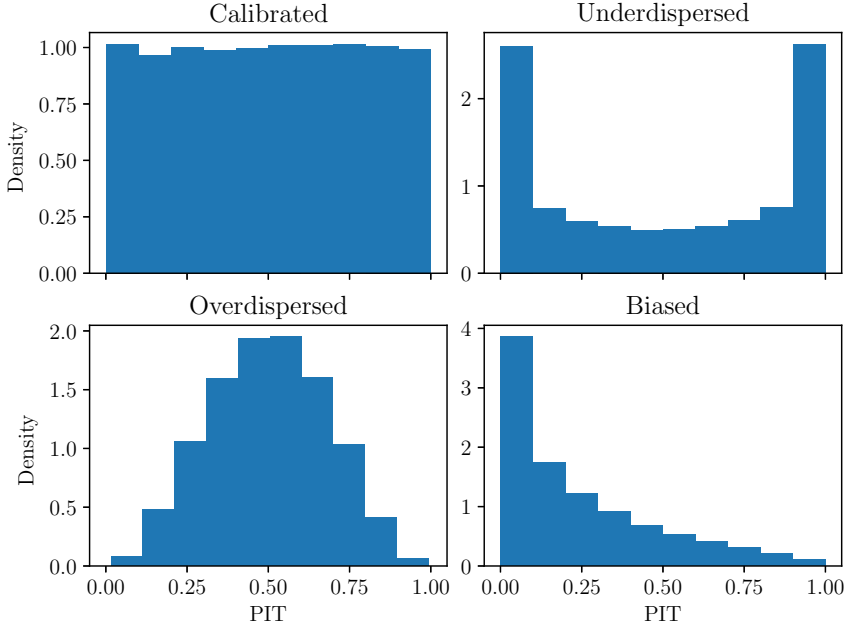
$$H(y) = \begin{cases} 0 & \text{if } y < 0, \\ 1 & \text{if } y \geq 0. \end{cases}$$

if conditional CDF is Heaviside step function (deterministic prediction $\hat{y}_i$) then the CRPS equals the *mean absolute error* (MAE):

$$\text{MAE} = |\hat{y}_i - y_i|$$

Evaluation: Probability integral transform (PIT) histogram

- evaluates the calibration of conditional CDFs
- histogram of values of $F(y_i|\mathbf{x}_i, \boldsymbol{\theta})$
- we can detect whether the model produces underdispersed, overdispersed, biased or calibrated conditional distributions



Multimodal regression

- $f_{Y_T|\mathbf{x}_T}(y|\mathbf{x})$ with several peaks
- deep ensemble's paper: "In cases where the Gaussian is too-restrictive, one could use a complex distribution e.g. mixture density network"

Mixture density network (MDN)

- model a conditional distribution $f_{\mathbf{X}_T|Y_T}(y|\mathbf{x})$
- using a mixture model for $f(y|\mathbf{x}_i, \boldsymbol{\theta})$
- sufficiently flexible network can approximate arbitrary conditional distributions

MDN: Gaussian components

model for Gaussian components (*heteroscedastic* model)

$$f(y|\mathbf{x}_i, \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k(\mathbf{x}_i) \mathcal{N}(y|\boldsymbol{\mu}_k(\mathbf{x}_i), \sigma_k^2(\mathbf{x}_i)).$$

MDN: Number of outputs

for K components in the mixture model the network will have

- K outputs denoted by a_k^{π} that determine $\pi_k(\mathbf{x}_i)$
- K outputs denoted a_k^{σ} that determine $\sigma_k(\mathbf{x}_i)$
- K outputs denoted by a_k^{μ} that determine $\mu_k(\mathbf{x}_i)$

MDN: Output layer

mixing coefficients must satisfy $\sum_{k=1}^{K} \pi_k(\mathbf{x}_i) = 1$ and $0 \leq \pi_k(\mathbf{x}_i) \leq 1$, therefore, softmax output

$$\pi_k(\mathbf{x}_i) = \frac{\exp(a_k^\pi)}{\sum_{j=1}^{K} \exp(a_j^\pi)}$$

variances must satisfy $\sigma_k^2(\mathbf{x}_i) \geq 0$, therefore exponentials $\sigma_k(\mathbf{x}_i) = \exp(a_k^\sigma)$ or the softplus function

means directly $\mu_k(\mathbf{x}_i) = a_k^\mu$

MDN: Illustration

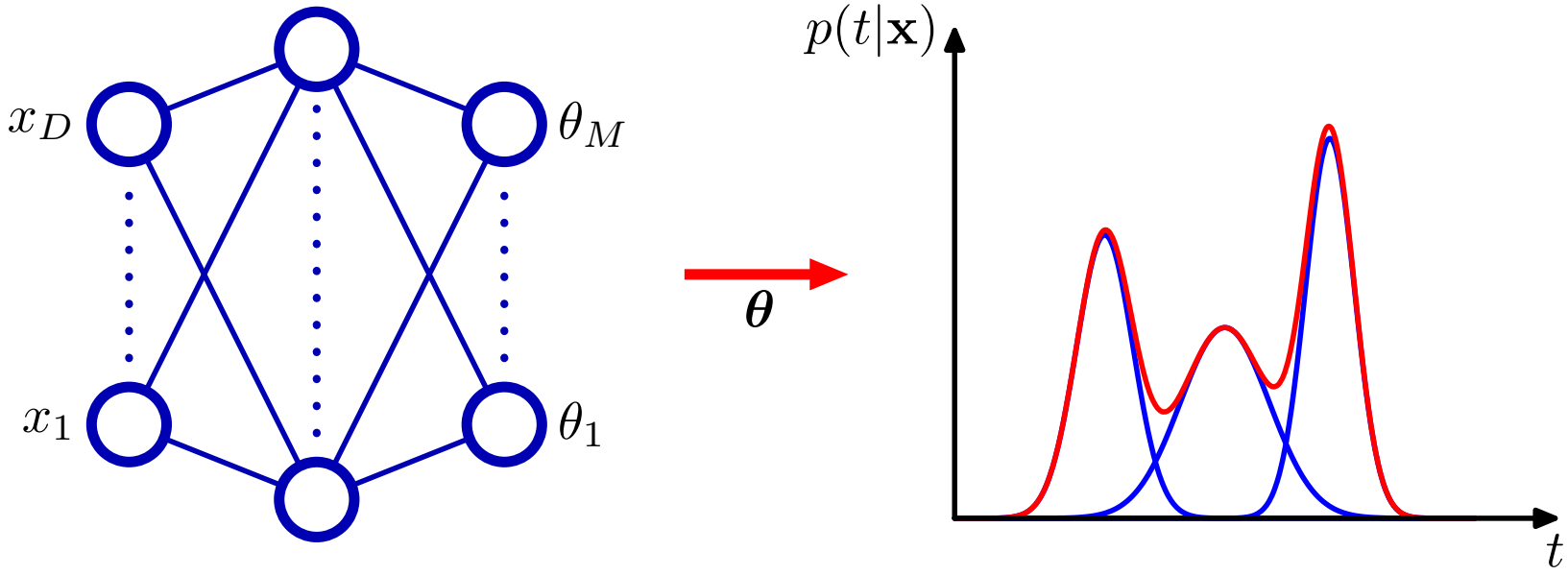


Figure: MDN outputs parameters of a parametric mixture model. Credits to Bishop (2006).

Future work

- active domain adaptation for multimodal regression
- use PIT histograms to identify multimodal problems
- decompose MDN predictive uncertainty into data, model and distributional uncertainty
- compare NLL and CRPS as loss functions