present

Active Learning: classical and deep approaches

Vaclav Smidl

AI tea, FEL ZCU,

29 March 2023

Motivation: Money, Money, Money!

Text Document clasification

Input : document files .txt

Output : document category, i.e. discrete label

Standard task in ML:

1. create embedding (e.g. Bert) for each document
2. train a classifier for the embedding, e.g. NN

Challenge:

- 1TB of documents of an international company suspected of a crime
- classify all documents into those relevant/irrelevant for the court
- a lawyer can judge the content for 500$/hour

Active learning

1. start with an initial batch of documents with labels, and the rest unlabeled
2. Select from unlabeled the `most interesting' documents for labeling
3. Obtain labels for the selected documents
4. Train Model,
5. GOTO 2

Many ways to define the `interesting' samples: heuristics × theoretically grounded.

Special Case of Decision making under uncertainty

Theory of optimal/rational decision making.

Decision : variable that we are free to select (document, experimental conditions)

Knowledge : data acquired so far (labeled documents, data), D

Uncertainty : outcome of the experiment (label)

Utility : What is the `useful' outcome of the experiment

Solution:

$$x^* = \arg\max_{x\in\mathcal{X}} \mathsf{E}_{y|D} U(y, D, x)$$

Cases:

1. *Bayesian optimization* : Minimization Utility + Gaussian Process
2. *Parametric Active Learning*: Mutual infomration + Parametric Model
3. *Deep Active Learning*: (motivated) heuristics

Special case #1: Bayesian optimization

Classical technology:

- Močkus, Jonas. *On Bayesian methods for seeking the extremum*. In Optimization Techniques IFIP Technical Conference: Novosibirsk, July 1–7, 1974, pp. 400-404. Springer Berlin Heidelberg, 1975.
  - first definition
- Jones, Donald R., Matthias Schonlau, and William J. Welch. *Efficient global optimization of expensive black-box functions.* Journal of Global optimization 13, no. 4 (1998): 455.
  - expected improvement
- Shahriari, Bobak, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando De Freitas. *Taking the human out of the loop: A review of Bayesian optimization.* Proceedings of the IEEE 104, no. 1 (2015): 148-175.
  - nice tutorial

Why to learn?

- (almost) all theoretical concepts have analytical solutions,
- useful for blackbox optimization, hyperparameter tuning,

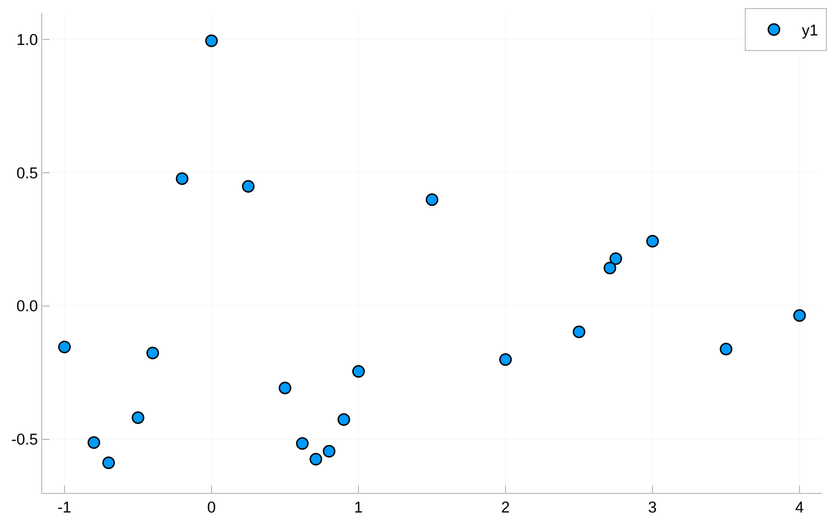
Human-based optimization

Find minimum of a function given by point-wise evaluation

```
x0=[-1.,0.,1.,2.,3.,4.];
```

Add points to x0 by interaction with audience...

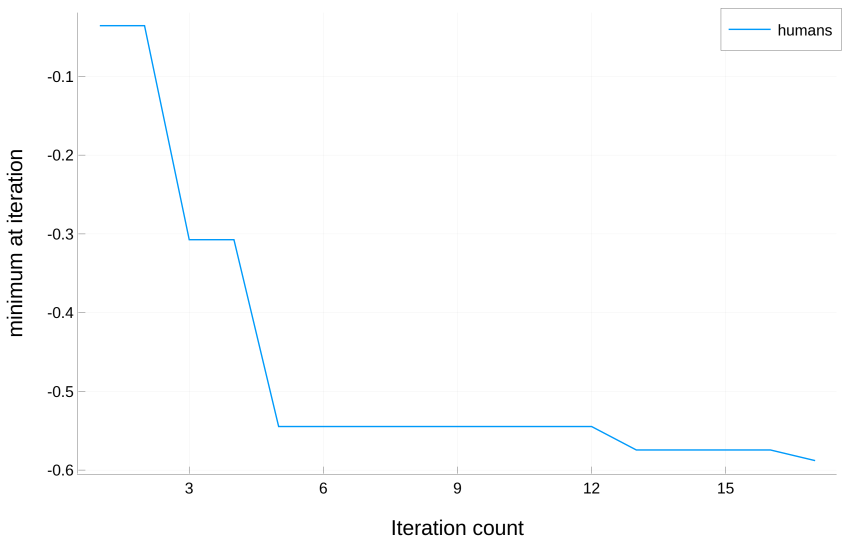
```
xh=[x0..., 1.5,0.5,0.25,0.8,0.9,-0.5,-0.2,2.5,0.618,2.75,2.71,0.71,3.5,-.4,-0.8,-0.7];
```




```
Plots.scatter(xh,f(xh),ylims=[-0.7,1.1])
```

How fast it was?

Evolution of the best minima as a function of iterations



Fast optimization as Decision making

Decision : variable that we are free to select, x

Knowledge : data acquired so far $D = [X, Y], X = [x_1, \ldots, x_n], Y = [y_1, \ldots, y_n]$,

Uncertainty : function value at x, $f(x)$

Utility : useful outcome is when $f(x) < \min(X)$

Solution:

$$x^* = \arg\max_{x \in \mathcal{X}} \int \chi(f(x) < \min(X)) p(f(x)|D) df(x)$$

Gaussian process: distribution over functions

A time continuous function $\{f(x); x \in \mathcal{X}\}$ is a Gaussian process if for every finite set of indeces $\{x_1, \ldots, x_n\}$ are Gaussian distributed, given functions

$$\begin{aligned} \textbf{mean:} \quad & \mu(x) \\ \textbf{kernel:} \quad & k(x, x') \end{aligned}$$

Consider function values $f(x), f(x')$, for two points x, x' with choices $\mu(x) = 0$, $k(x, x) = \nu$, then:

$$p(f(x), f(x')|\theta) = \mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}\nu+\sigma & k_\theta(x,x')\\ k_\theta(x,x') & \nu+\sigma\end{bmatrix}\right),$$

Gaussian distribution has very nice analytical properties:

$$\begin{aligned} p(f(x)|x, x_1, f(x_1)) &= \mathcal{N}\left(\mu_{x_1}(x), \sigma_{x_1}(x)\right), \\ \mu_{x_1}(x) &= k(x, x_1) f(x_1) \\ \sigma_{x_1}(x) &= (\nu + \sigma - k(x, x_1)^2/(\nu + \sigma)) \\ p(f(x)) &= \mathcal{N}(0, \nu) \end{aligned}$$

Read:

- Rasmussen, Carl Edward, and Christopher KI Williams. Gaussian processes for machine learning. Vol. 1. Cambridge, MA: MIT press, 2006.

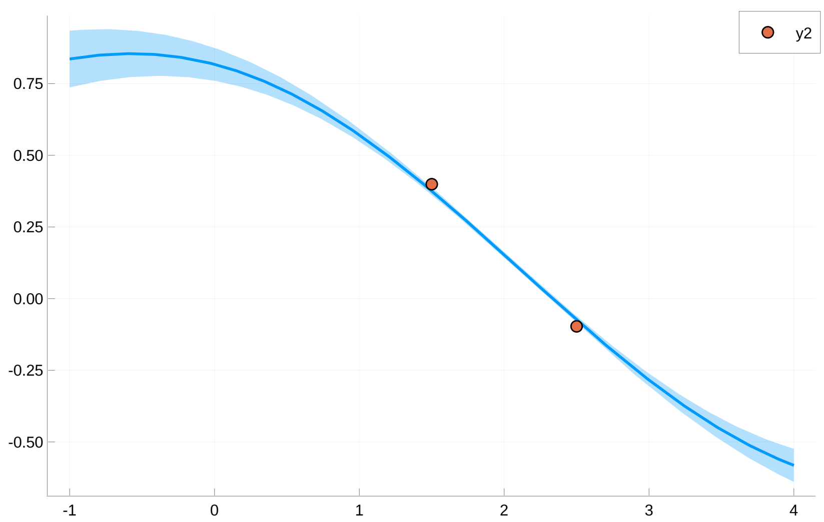
Gaussian Process Kernels

Sensitivity to hyper-parameters:

- kernel $K()$: KernelFunctions.SqExponentialKernel
- prior scale $\nu =$:
- length-scale $\ell =$:
- noise variance $\sigma =$:

```
begin xt=[1.5,2.5]; yt=f(xt); end;
```

```
testGP=buildgp(xt,[ν,ℓ,σ],Kf);
```




```
plotBO(posterior(testGP,yt),xt,false)
```

Estimating hyper-parameters

The fit before was done for known valued of hyperparameters θ, i.e.

$$p(f(x)|x_1, y_1, x, \theta)$$

Since it is a proper likelihood, we can compute best parameter value $\hat{\theta}$ for given points

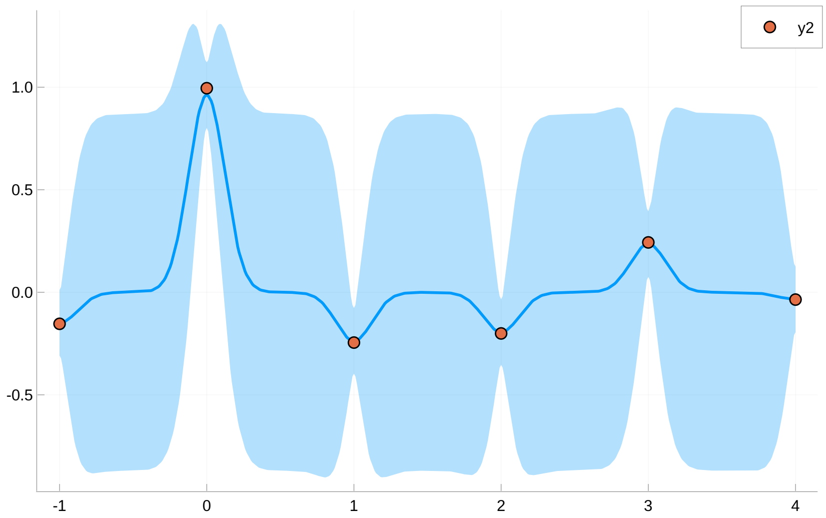
$$\hat{\theta} = \arg\min - \log p(Y|X, \theta)$$

where $X = [x_1, \ldots, x_n], Y = [y_1, \ldots, y_n]$

- poor but simple!

```
θ0 = [1.0,1.,-5];
```

```
opt = Optim.optimize(p->-logpdf(buildgp(x0,p),f(x0)), θ0, LBFGS());
```



Where it is most likely that the function has a minimum?

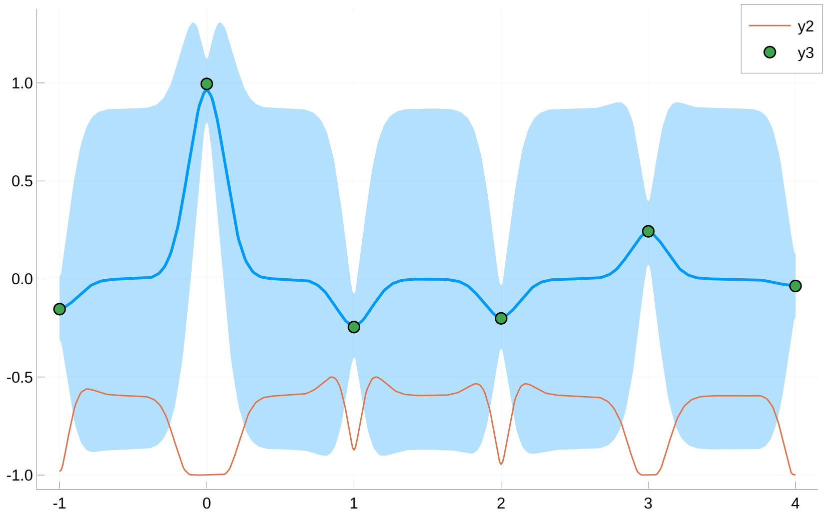
In each point x' the function value $f(x')$ has a Gaussian distribution

$$f(x) \sim G(\mu(x), \sigma(x))$$

Probability of a value being lower than a threshold is an analytical formula:

$$\begin{aligned} P(f(x) < \min(X)|Y, X) &= \int_{-\inf}^{\min(X)} G(\mu_X(x), \sigma_X(x)) df(x) \\ &= (\min(X) - \mu_X(x))\mathrm{cdf}(Z). + \sigma_X(x)\mathrm{pdf}(Z) \\ Z &= (\min(X) - \mu_X(x))/\sigma_X(x) \end{aligned}$$

```
EI (generic function with 1 method)
```

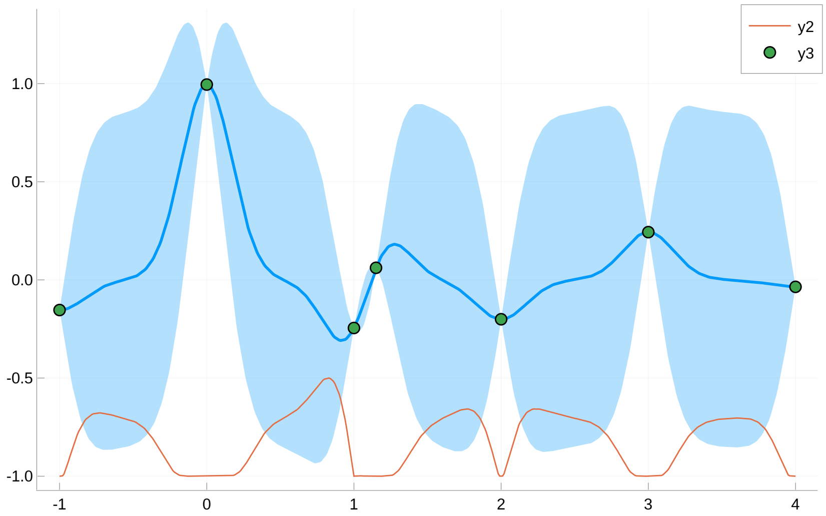


- Evaluate the point with highest probability of occurence of lower value.

Repeating the procedure for each point

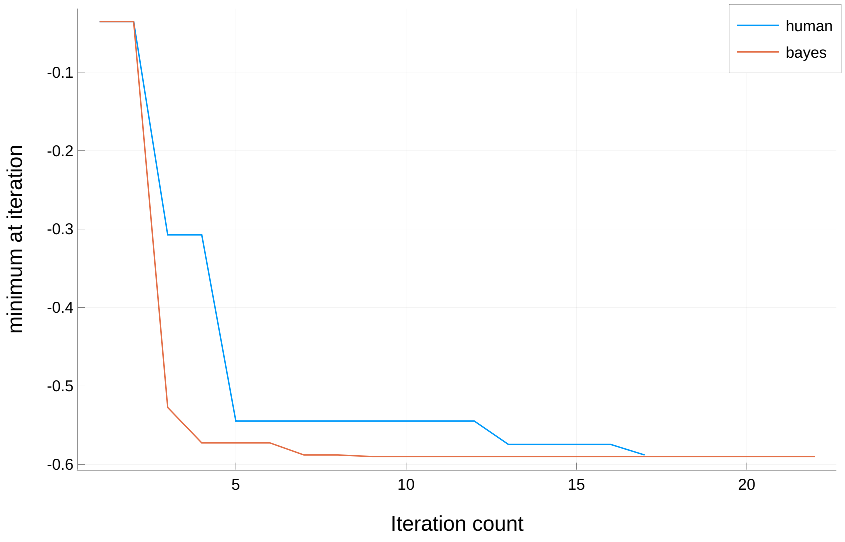
```
nextx (generic function with 2 methods)
```

Step of GP:




```
plotBO(G[st],X[st-1])
```

Is AI smarter than us?



Special Case #2: Active Learning with parametric models

The goal in not to find a good model with as few data as possible.

- maximum likelihood will not help (asymptotics)
- parameter estimates `compress' information from the data

$$p(\theta|X,Y) \propto p(Y|X,\theta)p(\theta)$$

- the more data I have, the more narrow the the parameter estimates are (if the data are informative!)

Decision making concepts:

Uncertainty: model parameters θ, can include discrete (order!)

Utility: new observation is useful if it improves parameter estimates

The goal si to gain information

Decision making task where utility is the prior-posterior gain in Shannon information is the *mutual information* between new observation $I(\theta; y)$

$$\begin{aligned} U(x) = E_y(I(\theta;y)) &= \int_y I(\theta;y) \\ &= -\int\int \log(p(\theta|y,x,D))\,p(\theta,y|x,D)\,d\theta\,dy + \int \log(p(\theta,D))\,p(\theta|D)\,d\theta \\ &= -\int\int \log(p(y|\theta,x,D))\,p(\theta,y|x,D)\,dy\,d\theta + \int \log(p(y|x,D))\,p(y|x,D)\,dy, \\ p(y|x,D) &= \int p(y|x,\theta,D)d\theta \end{aligned}$$

Theoretically nice, practially intractable for interesting problems.

Approximations to the rescue:

- entropy of prediction
  - denoted $H(y|x,\hat{\theta}) = -\int \log p(y|x,D,\hat{\theta})p(y|x,D,\hat{\theta})dy$
  - for Gaussian distribution, $H(y) = 0.5\log(2\pi\sigma^2) + 0.5$
- for sampled parameters (ensembles)

$$p(\theta) = \{\theta_1 \ldots \theta_n\}$$
$$p(y|x) = 1/n \sum_i p(y|x,\theta_i)$$
$$U(x) = 1/n \sum_i H(y|x,\theta_i) - H(y|x)$$

- can be evaluated efficienty : note that the parameters are the same for all considered samples!

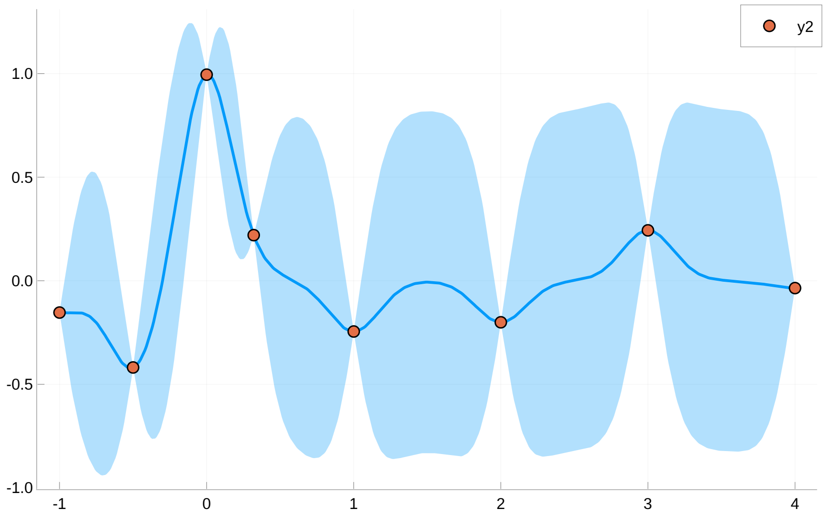
How it works for previous example?

Same model: GP

Different utility: Mutual Information instead of expected improvement

```
nextxv (generic function with 2 methods)
```

Step of GP:



Why it is just grid-refining?

- answers is in the assumptions
-
-

Kernel is the same for all points!

Learning with parametric models

Consider situation that we know the model but not its the parameters

$$f(x) = \frac{\theta_3 \cos(\theta_1 x + \theta_2)}{\theta_5 x^2 + |x| + \theta_4}$$

```
f (generic function with 2 methods)
```

```julia
f(x,θ)=θ[3]*cos.(θ[1]*x.+θ[2])./(θ[5]*x.^2+abs.(x).+θ[4])
```

```
activef (generic function with 2 methods)
```

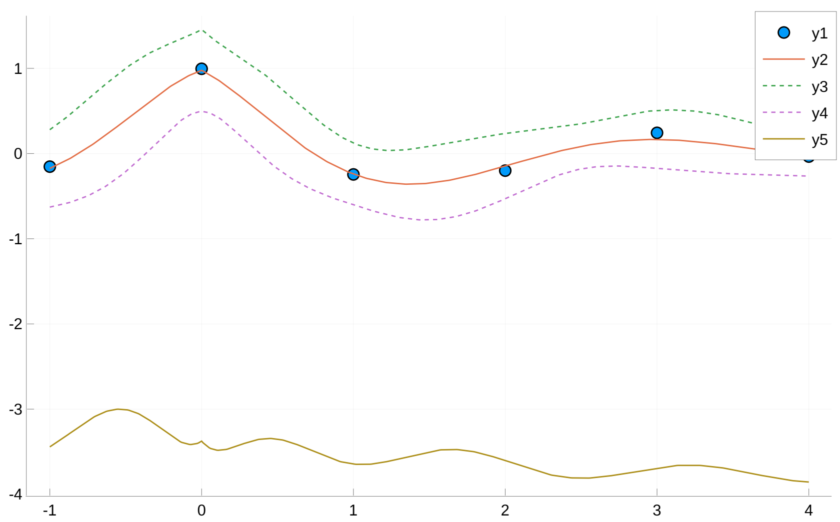
```julia
# write it as a probabilistic program
@model function activef(x, y)
  θ ~ MvNormal([4.3,0.0,1.0,1.0,0.0], I)   # Prior

  for i in eachindex(x)                     # Observations
    y[i] ~ Normal(f(x[i],θ), 0.1)
  end
end
```

```julia
model = activef(x0,f(x0));
```

```julia
Θ=sample(model, NUTS(), MCMCThreads(), 1000, 1);
```

```
100%
Found initial step size
ϵ: 0.4
```

```
plotParam (generic function with 1 method)
```




```julia
plotParam(x0,f(x0),allY)
```

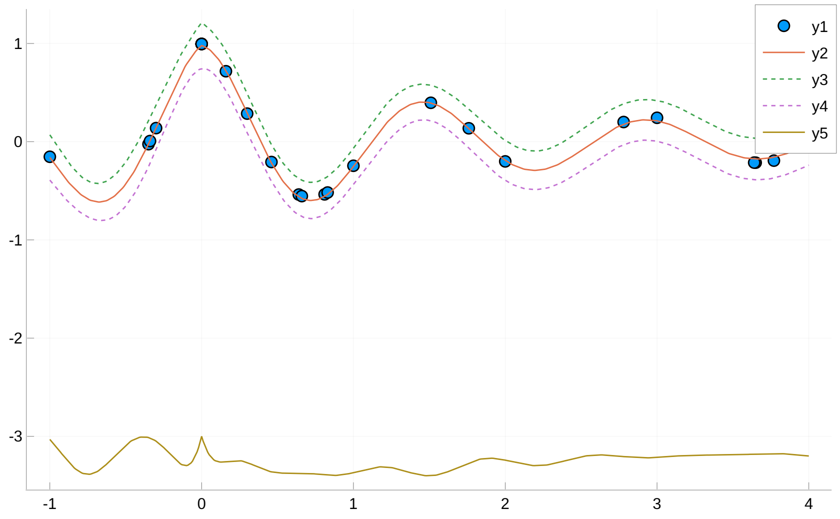
```
nextxm (generic function with 1 method)
```

```julia
function nextxm(x)
    model = activef(x,f(x));
    Θ=sample(model, NUTS(), MCMCThreads(), 1000, 1)
    xg  = -1:0.01:4
    allY=[f(xg,Θ.value.data[i,1:5,1]) for i=1:1000]
    vY=var(allY);
    xg[argmax(vY)],allY
end
```

```
100%
Found initial step size
ϵ: 0.00625
```

```
100%
Found initial step size
ϵ: 0.4
100%
Found initial step size
ϵ: 0.2
100%
Found initial step size
ϵ: 0.00625
100%
Found initial step size
ϵ: 0.025
100%
Found initial step size
ϵ: 0.2
100%
Found initial step size
ϵ: 0.0015625
100%
Found initial step size
ϵ: 0.025
100%
Found initial step size
ϵ: 0.025
100%
Found initial step size
ϵ: 0.025
100%
```

Step of GP:




```
plotParam(Xm[stm],f(Xm[stm]),AY[stm])
```

Special case #3: Active learning with Neural Networks

Uncertainty : output of neural network for given input

Utility : Approximation of Mutual information

Approximations - almost heuristics

Classifiers are often trained using the `crossentropy' loss

- it is exactly the conditional approximation of predictive entropy !

If we want better uncertainty? Use ensembles

- Dropout MC is a strategy sampling dropout variables even in prediction
- Deep Ensembles = train NN from different initial conditions
    - inefficient for changing only a single value
    - warm start: initialize new weight by random perturbation of previous

How about Utility function:

- Entropy: sum-of-entropy — entropy-of-sum
- BALD: Houlsby, Neil, Huszar, Ferenc, Ghahramani, Zoubin, and Lengyel, Mate. Bayesian active learning for classification and preference learning. arXiv preprint arXiv:1112.5745,2011.

Active Learning for Document Classification

Benchmark data:

- Tweets Dataset
- News Category Dataset
- Fake News,
- Fake News Detection

Embeddings:

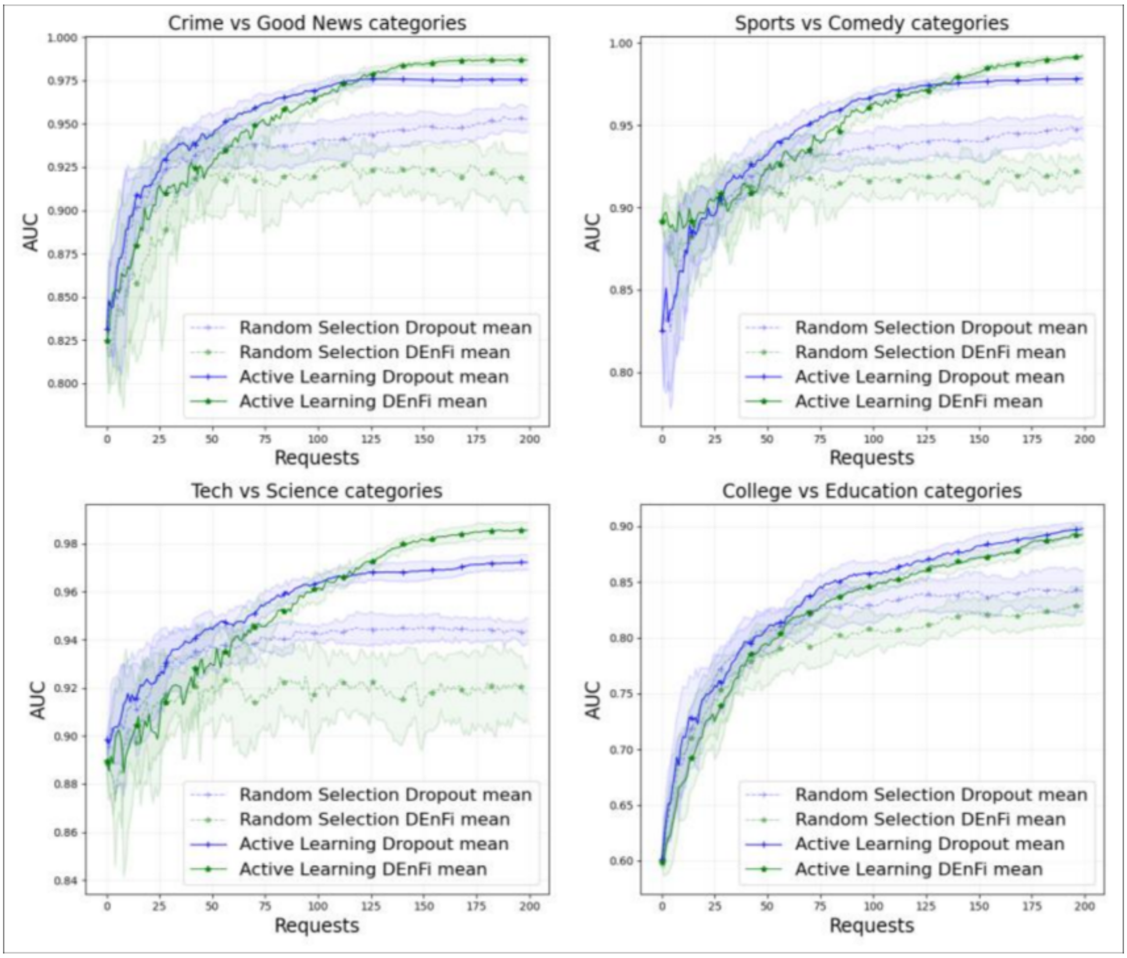
- Fast Text
- LASER
- RoBERTa

More details:

- Sahan, Marko, Vaclav Smidl, and Radek Marik. *Active learning for text classification and fake news detection.* In 2021 International Symposium on Computer Science and Intelligent Controls (ISCSIC), pp. 87-94. IEEE, 2021.

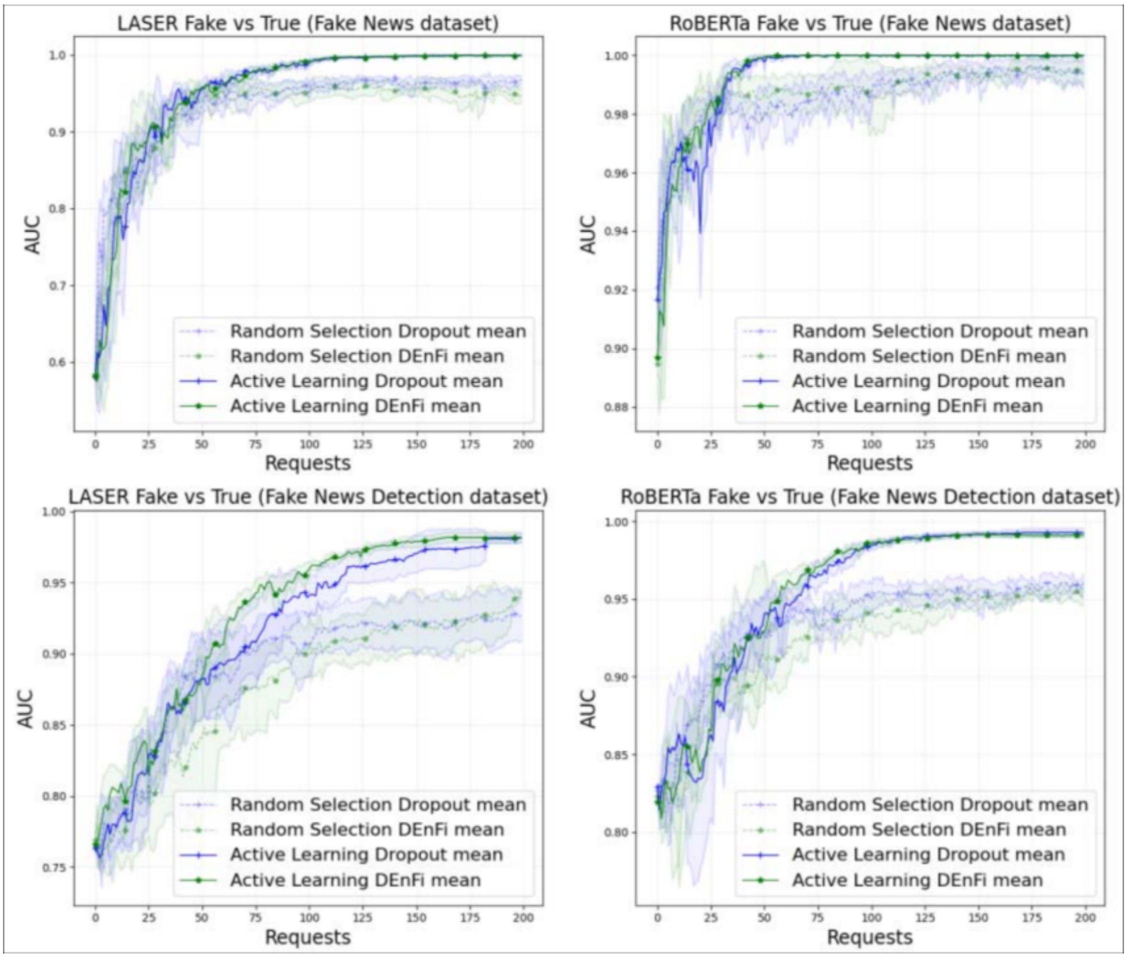
Simple vs. complex models: Exploration-exploitation tradeoff

L =



Sensitivity to document embedding

Le =



Taking Human out of the loop

People can not be part of the training loop, constantly waiting for new assignmets.

Prefer to process *batches* of, say 10, documents.

Optimal batch selection:

- much harder job
- selection of a single sample is easy
- selection of a pair is harder: n^2 possibilities!

Fear:

- selecting 10 best independently has low value

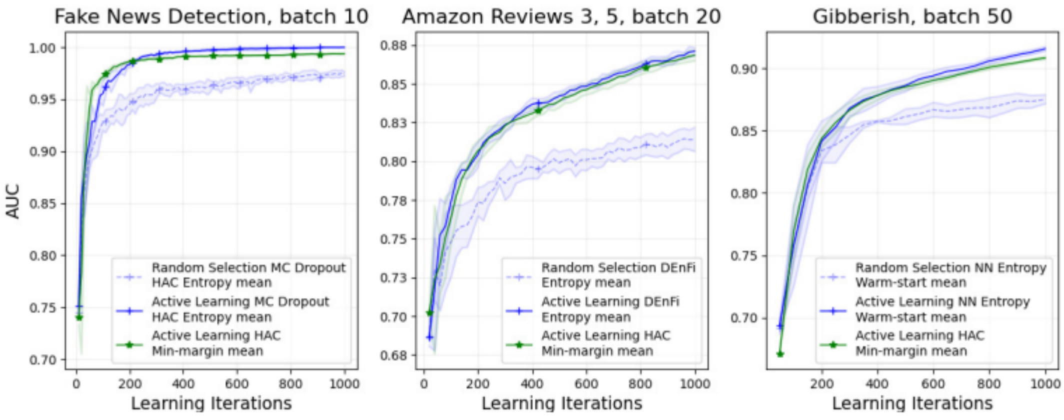
Heuristics:

- cluster the samples, select representant of each cluster (HAC)

Comparison on Document classification

- Sahan, Marko, Vaclav Smidl, and Radek Marik. *Batch Active Learning for Text Classification and Sentiment Analysis.* In Proceedings of the 2022 3rd International Conference on Control, Robotics and Intelligent System, pp. 111-116. 2022.
- Slower learning than with one sample loop
- Overall best: single network with cross-entropy and cold/Warm-start after each acquisition batch
  - the documents are diverse enough ?
- Ensembles/Dropout MC useful for the fake news dataset

L2 =




```
L2=load("fake2.png")
```

Take home message

- active learning is a way to to reduce cost of training
    - can be also applied to testing
- Danger:
    - non i.i.d. sampling
    - being stuck in local extreme
- Remedies:
    - use stochastcicity (warm starts, reinitializations)
    - you may waste a few samples but gain information
- Active line of our research - finetuning to their specifics:
    - controller tuning
    - design optimization
    - material experiment design
    - plasma physics experiments/simulations
- New application?