Tools for hyperparameter tuning

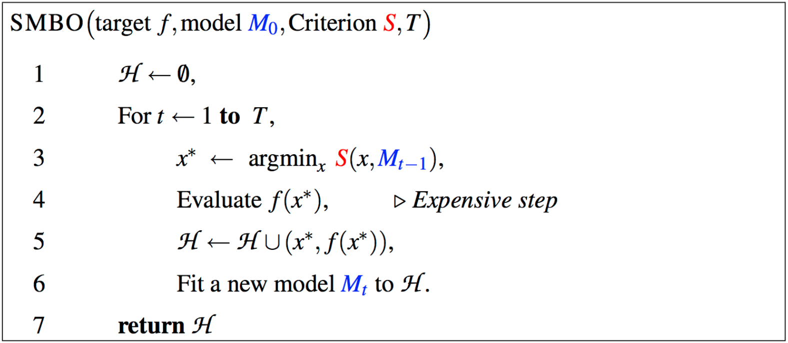
Siminole meeting

Matthias Brendel and Rémi Bardenet

LAL, LRI, Univ. Paris-Sud XI

January 25th 2012

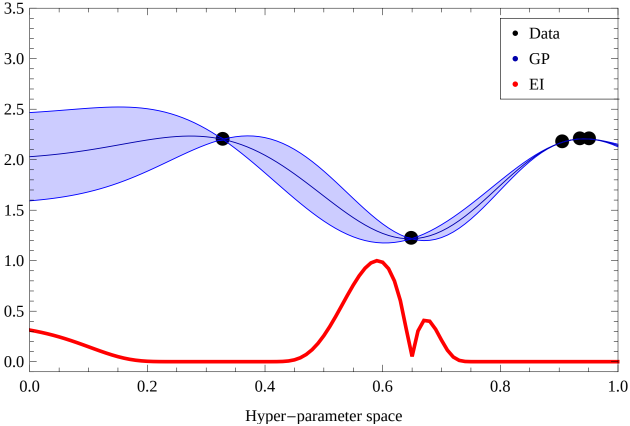
Sequential model-based optimization



```
SMBO(target $f$, model $M_0$, Criterion $S$, $T$)
    $\mathcal{H} \leftarrow \emptyset$,
    For $t \leftarrow 1$ **to** $T$,
        $x^* \leftarrow \operatorname{argmin}_x S(x, M_{t-1})$,
        Evaluate $f(x^*)$,        ▷ *Expensive step*
        $\mathcal{H} \leftarrow \mathcal{H} \cup (x^*, f(x^*))$,
        Fit a new model $M_t$ to $\mathcal{H}$.
    **return** $\mathcal{H}$
```

- Especially useful when target evaluation is costly.

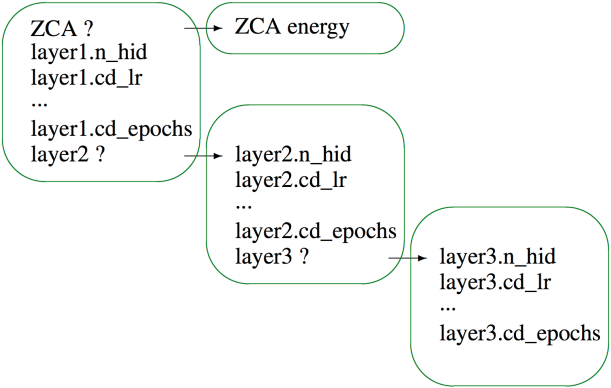
Gaussian Processes and Expected Improvement



- GPs are priors over functions that are **closed under sampling**.
- EI$(x) := \mathbb{E}\big((\min_i f(x_i) - f(x)) \wedge 0 | \mathcal{F}_n\big)$.

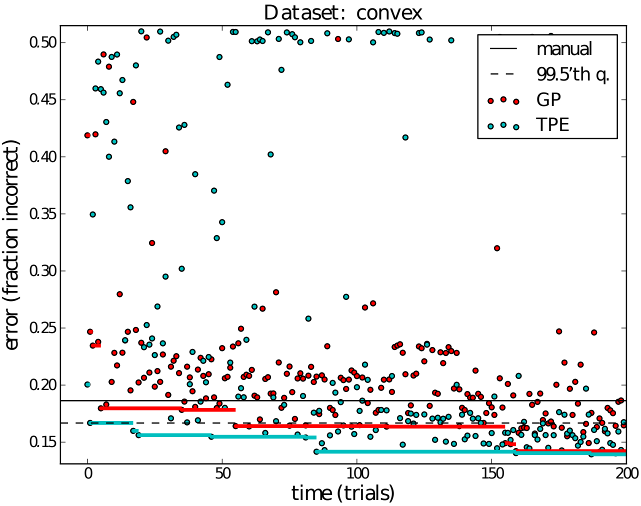
A recent application to hyperparameter tuning

- Deep Belief Nets have **lots of conditional hyperparameters**,



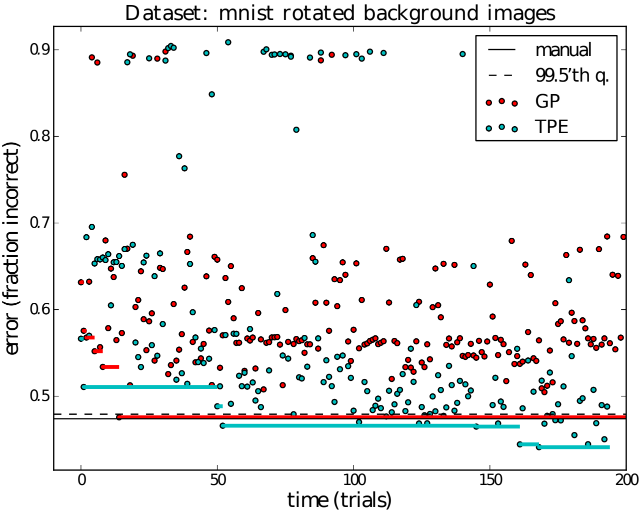
A recent application to hyperparameter tuning

- We used SMBO, using GPs+EI and a tree-based model+EI in Bergstra, Bardenet, Kégl and Bengio, NIPS'11.



A recent application to hyperparameter tuning

- We used SMBO, using GPs+EI and a tree-based model+EI in Bergstra, Bardenet, Kégl and Bengio, NIPS'11.



{Learning from, tuning for} different datasets

- As of today, what we call a problem is actually a dataset, with a certain number of features.
- Idea is to place a GP over an **augmented feature+hyperparameters space**.
- But error rates coming from different datasets are not comparable!
- Then a GP over the error rate function is unrelevant.

{Learning from, tuning for} different datasets

- As of today, what we call a problem is actually a dataset, with a certain number of features.
- Idea is to place a GP over an **augmented feature+hyperparameters space**.
- But error rates coming from different datasets are not comparable!
- Then a GP over the error rate function is unrelevant.

Idea

1. Store the pairwise rankings given by the evaluation of your algorithm on single datasets.
2. Infer a **flat latent function** that preserves ranking:

$$u \prec v \Leftrightarrow \ell(u) < \ell(v).$$

An existing GP+ranking framework

- GPs need to be tuned.
- Usually, it's done by maximizing the marginal likelihood of the hyperparameters of the GP.
- This approach is unrelevant here, as one does **not even know the values** of the latent function.
- Chu and Gharamani, NIPS'05 proposed an algorithm that takes as input the pairwise rankings and that **simultaneously**
  - estimate the ranking-preserving latent function,
  - and tune a GP placed over it.
- Very expensive. Replaced by SVMrank (Joachims, '02) on our preliminary experiments.

Thanks for your attention. Now, back to Matthias!