Scalable Thompson Sampling for Non-Conjugate Models

James McInerney

Netflix

jmcinerney@netflix.com

Bayesian Causal Inference for Real World Interactive Systems

August 15, 2021

Exploration in Recommendation

Collaborative Filtering

out of matrix prediction

X : context

A : recommendation

			
	0.1	0.3	?
	0.3	0.05	?
	?	?	?

cold start

non-stationarity

Thompson Sampling

Given dataset $\mathcal{D}$, likelihood $p(\mathcal{D} \mid \theta)$, and prior $p(\theta)$ with model variables θ.

Iterate over the following two steps:

1. Sample model variables from posterior $\theta^{(s)} \sim p(\theta \mid \mathcal{D})$
2. Pick action $a^* = \arg_a \max \mathbb{E}[\mathrm{reward}(a) \mid \theta^{(s)}]$

Thompson Sampling

Given dataset $\mathcal{D}$, likelihood $p(\mathcal{D} \mid \theta)$, and prior $p(\theta)$ with model variables θ.

Iterate over the following two steps:

1. Sample model variables from posterior $\theta^{(s)} \sim p(\theta \mid \mathcal{D})$
2. Pick action $a^* = \arg_a \max \mathbb{E}[\text{reward}(a) \mid \theta^{(s)}]$

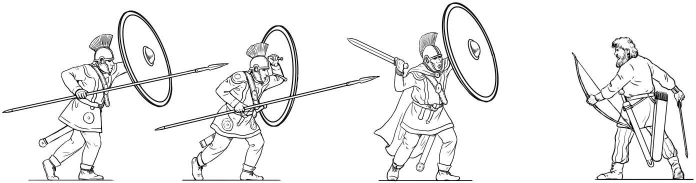
Unless $p(\mathcal{D} \mid \theta)$ and $p(\theta)$ take the same form (conjugate), the posterior calculation is computationally prohibitive for many models.

Exploration for Non-Conjugate Models

- Complex models (e.g. deep neural nets, gradient boosted decision trees) present challenges for quantifying uncertainty.
- Hard to apply existing exploration-exploitation schemes (e.g. UCB, Thompson sampling) off the shelf.

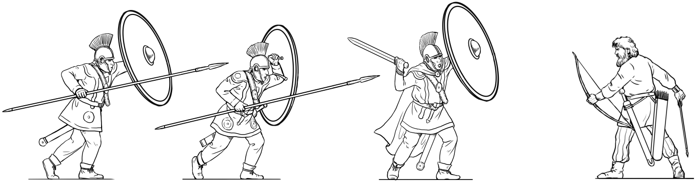
no/low model uncertainty	**model uncertainty**	
	bootstrapping MCMC	**computationally expensive** (relative to base model)
epsilon-greedy Boltzmann exploration last layer variance	model-specific approximations (e.g. Laplace) residual overfit (this talk)	**computationally inexpensive** (relative to base model)

On Avoiding Overfitting



regularization dropout model selection overfitting

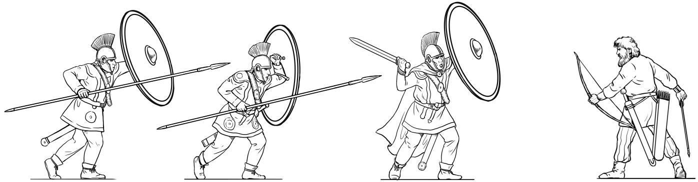
On Avoiding Overfitting



regularization dropout model selection overfitting

- Amazing literature on how to defend against overfitting.

On Avoiding Overfitting



- Amazing literature on how to defend against overfitting.
- Is overfitting ever useful?

The Residual Overfit

- Let $f : \mathcal{X} \to \mathbb{R}$ be a regularized model and $g : \mathcal{X} \to \mathbb{R}$ be an unregularized model that maximally overfits the training data.

The Residual Overfit

- Let $f : \mathcal{X} \to \mathbb{R}$ be a regularized model and $g : \mathcal{X} \to \mathbb{R}$ be an unregularized model that maximally overfits the training data.

- Formally, if $h(x)$ is the unknown true function, then assume:

  $f(x) = h(x) + \beta$, for any $\beta \perp\!\!\!\perp \epsilon$
  $g(x) = h(x) + \epsilon$, where $\mathbb{E}[\epsilon] = 0$

  where $\mathbb{E}[\cdot]$ is with respect to the true population distribution.

The Residual Overfit

- Let $f : \mathcal{X} \to \mathbb{R}$ be a regularized model and $g : \mathcal{X} \to \mathbb{R}$ be an unregularized model that maximally overfits the training data.

- Formally, if $h(x)$ is the unknown true function, then assume:

  $f(x) = h(x) + \beta$, for any $\beta \perp\!\!\!\perp \epsilon$
  $g(x) = h(x) + \epsilon$, where $\mathbb{E}[\epsilon] = 0$

  where $\mathbb{E}[\cdot]$ is with respect to the true population distribution.

- We introduce the concept of the **residual overfit** defined as:

  $\text{residual overfit} = |f(x) - g(x)|.$

The Residual Overfit

- Let $f : \mathcal{X} \to \mathbb{R}$ be a regularized model and $g : \mathcal{X} \to \mathbb{R}$ be an unregularized model that maximally overfits the training data.

- Formally, if $h(x)$ is the unknown true function, then assume:

  $f(x) = h(x) + \beta$, for any $\beta \perp\!\!\!\perp \epsilon$
  $g(x) = h(x) + \epsilon$, where $\mathbb{E}[\epsilon] = 0$

  where $\mathbb{E}[\cdot]$ is with respect to the true population distribution.

- We introduce the concept of the **residual overfit** defined as:

  $\text{residual overfit} = |f(x) - g(x)|.$

- Claim: the squared residual overfit is an upper bound on the error of $f(x)$ with respect to the true function $h(x)$.

Residual Overfit Derivation # 1

Train f and g on two distinct random splits of datasets drawn from the true population distribution F. Calculate residual overfit on query point x,

$$
\begin{aligned}
&\mathbb{E}[(f(x) - g(x))^2] \\
&= \mathbb{E}[(h(x) + \beta - h(x) - \epsilon)^2] \\
&= \mathbb{E}[(\beta - \epsilon)^2] \\
&= \mathbb{E}[\beta^2 - 2\beta\epsilon + \epsilon^2] \\
&= \mathbb{E}[\beta^2] + \mathbb{E}[\epsilon^2] \\
&= \mathrm{MSE}[f(x)] + \mathrm{Var}[g(x)]
\end{aligned}
$$

Residual Overfit Derivation # 1

Train f and g on two distinct random splits of datasets drawn from the true population distribution F. Calculate residual overfit on query point x,

$$
\begin{aligned}
&\mathbb{E}[(f(x) - g(x))^2] \\
&= \mathbb{E}[(h(x) + \beta - h(x) - \epsilon)^2] \\
&= \mathbb{E}[(\beta - \epsilon)^2] \\
&= \mathbb{E}[\beta^2 - 2\beta\epsilon + \epsilon^2] \\
&= \mathbb{E}[\beta^2] + \mathbb{E}[\epsilon^2] \\
&= \mathrm{MSE}[f(x)] + \mathrm{Var}[g(x)]
\end{aligned}
$$

Reminder: argument holds in expectation and needs $\beta \perp\!\!\!\perp \epsilon$.

Residual Overfit Derivation #2

Population averaged maximum *a posteriori* objective:

$$\mathcal{L}_\theta = \mathbb{E}_F[\log p(\mathcal{D} \mid \theta)] + \log p(\theta \mid \alpha)$$
$$\theta^* = \arg_\theta \max \mathcal{L}_\theta$$

Residual Overfit Derivation #2

Population averaged maximum *a posteriori* objective:

$$\mathcal{L}_\theta = \mathbb{E}_F[\log p(\mathcal{D} \mid \theta)] + \log p(\theta \mid \alpha)$$
$$\theta^* = \arg_\theta \max \mathcal{L}_\theta$$

Apply implicit function theorem:

$$\frac{\mathrm{d}\theta^*}{\mathrm{d}\alpha} = -\left(\mathbb{E}_F\left[\frac{\partial^2}{\partial\theta_i \partial\theta_j} \log p(\mathcal{D} \mid \theta)\right]\right)^{-1} \frac{\partial^2}{\partial\theta\partial\alpha} \log p(\theta \mid \alpha) \qquad (1)$$

Residual Overfit Derivation #2

Population averaged maximum *a posteriori* objective:

$$\mathcal{L}_\theta = \mathbb{E}_F[\log p(\mathcal{D} \mid \theta)] + \log p(\theta \mid \alpha)$$
$$\theta^* = \arg_\theta \max \mathcal{L}_\theta$$

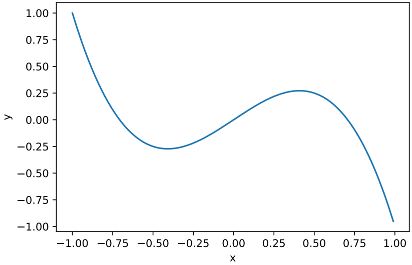
Apply implicit function theorem:

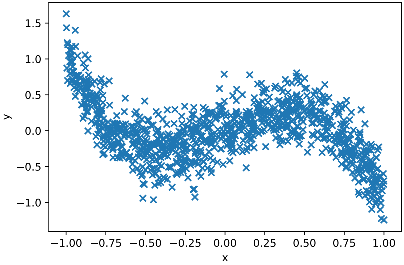
$$\frac{\mathrm{d}\theta^*}{\mathrm{d}\alpha} = -\left(\mathbb{E}_F\left[\frac{\partial^2}{\partial\theta_i\partial\theta_j}\log p(\mathcal{D} \mid \theta)\right]\right)^{-1}\frac{\partial^2}{\partial\theta\partial\alpha}\log p(\theta \mid \alpha) \qquad (1)$$

Residual overfit is a first-order Taylor-based approximation of Eq. 1.

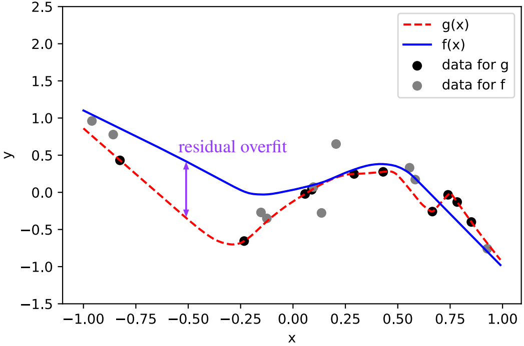
Synthetic Example

- True reward is $h(x) = x(1 - 2x^2)$.
- Continuous action space with two local optima in range [-1, 1].



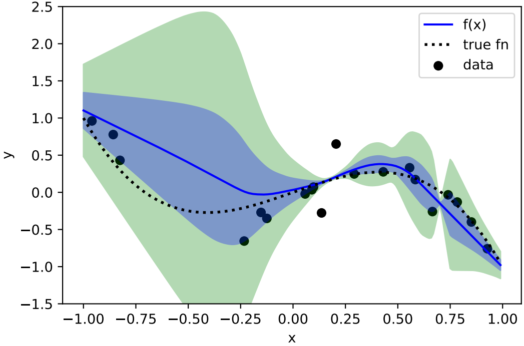


Synthetic Example



- Model architecture: 3-layer neural net each with 500 ReLu nodes.
- f: train for 20 epochs (early stopping).
- g: train for 1000 epochs (overfit).

Synthetic Example



- Inner blue shaded region: *residual overfit*.
- Outer green shaded region: 2.58 times residual overfit (99% confidence interval).

Why Not Just Fit to RMSE?

- Observed RMSE combines epistemic with aleatoric uncertainty.
- Example: generated 100 data points at input points $x \in \{-1, -\frac{1}{2}, 0, \frac{1}{2}, 1\}$.

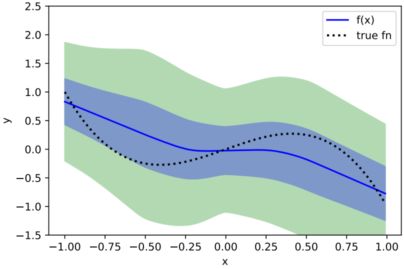


Figure: Fit model to observed RMSE

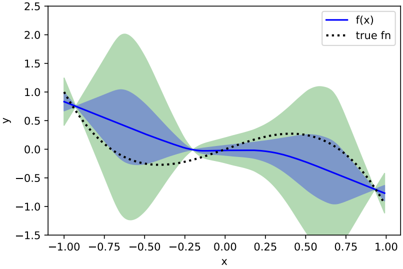
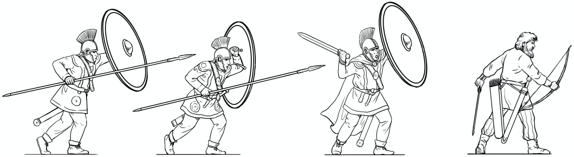
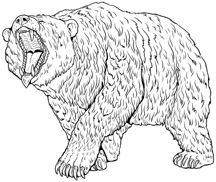


Figure: Residual overfit

Residual Overfit Method of Exploration (ROME)

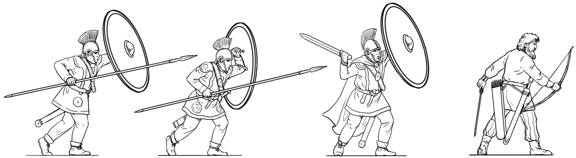


regularization   dropout   model selection   overfitting

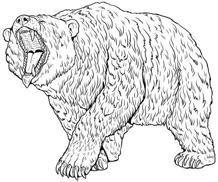


cold start

Residual Overfit Method of Exploration (ROME)



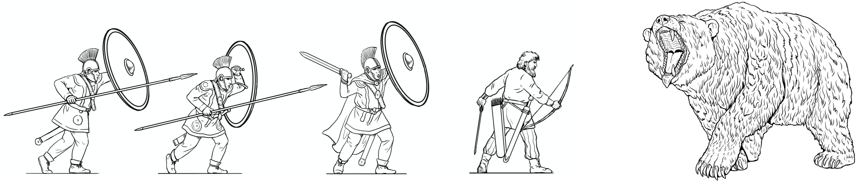
regularization dropout model selection overfitting



cold start

Can we use residual overfit for exploration?

Residual Overfit Method of Exploration (ROME)

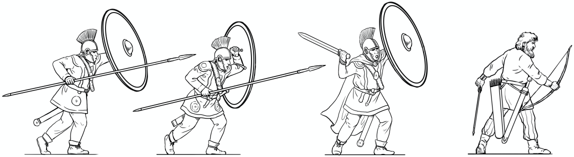


regularization dropout model selection overfitting cold start

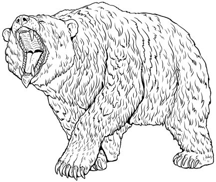
Can we use residual overfit for exploration?

UCB: $\text{reward}(x_a) = f(x_a) + \alpha|f(x_a) - g(x_a)|$, for hyperparameter α.

Residual Overfit Method of Exploration (ROME)



regularization dropout model selection overfitting



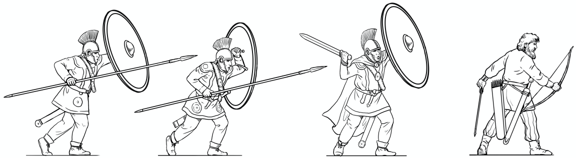
cold start

Can we use residual overfit for exploration?

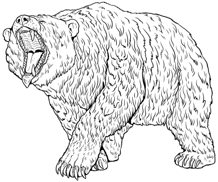
UCB: $\mathrm{reward}(x_a) = f(x_a) + \alpha|f(x_a) - g(x_a)|$, for hyperparameter α.

Thompson Sampling: $\mathrm{reward}(x_a) \sim \mathcal{N}(f(x_a), (f(x_a) - g(x_a))^2)$

Residual Overfit Method of Exploration (Rome)



regularization dropout model selection overfitting



cold start

Can we use residual overfit for exploration?

UCB: $\mathrm{reward}(x_a) = f(x_a) + \alpha|f(x_a) - g(x_a)|$, for hyperparameter α.

Thompson Sampling: $\mathrm{reward}(x_a) \sim \mathcal{N}(f(x_a), (f(x_a) - g(x_a))^2)$

Exponential family moment matching for different observation models.
E.g., for binary rewards, closed-form expression for the beta posterior.

Empirical Evaluation

- Base model: random forest classifier (RF).

Methods:

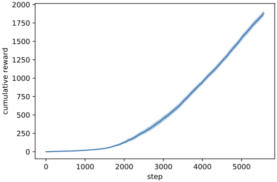
- Uniform random.
- Epsilon greedy, with $\epsilon = 0.1$.
- LinUCB, with $\alpha = 1.0$.
- Bootstrap Thompson sampling, with $M = 20$ bootstraps of the data.
- Rome-UCB & Rome-TS:
  - f: base model
  - g: a single decision tree
  - both f and g trained on the same data

Results

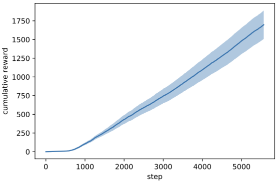
Table: Average regret with 95% confidence interval over 10 replications. Top performing method for each dataset in bold.

Method	Covertype (7 classes)	Bach Chorales (65 classes)	MovieLens (3600 classes)
LinUCB	0.415 ± 0.003	0.664 ± 0.007	0.967 ± 0.005
Epsilon Greedy	0.403 ± 0.005	0.711 ± 0.051	0.970 ± 0.000
Bootstrap-TS	$\mathbf{0.390 \pm 0.003}$	0.668 ± 0.028	0.971 ± 0.000
ROME-UCB	0.422 ± 0.004	0.718 ± 0.035	0.963 ± 0.005
ROME-TS	0.524 ± 0.004	$\mathbf{0.657 \pm 0.012}$	$\mathbf{0.941 \pm 0.006}$
Uniform Random	0.859 ± 0.004	0.985 ± 0.001	0.986 ± 0.001

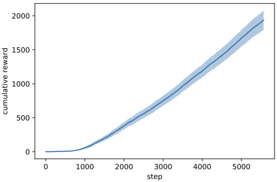
Bach Chorales



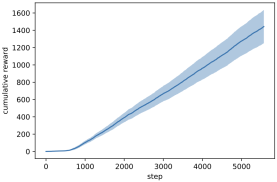
(a) LinUCB



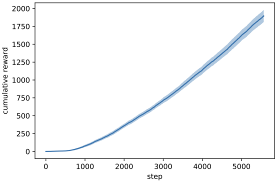
(b) Epsilon Greedy



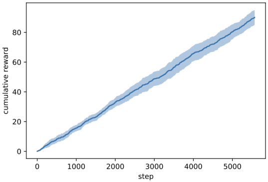
(c) Bootstrap-TS



(d) Rome-UCB

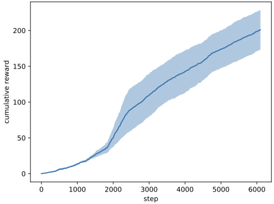


(e) Rome-TS

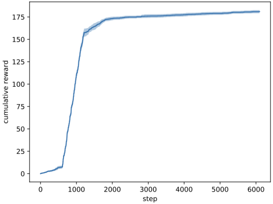


(f) Uniform Random

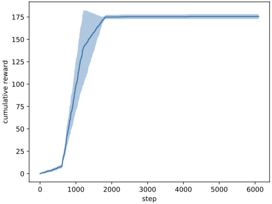
MovieLens (Depleting Actions)



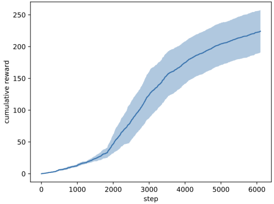
(a) LinUCB



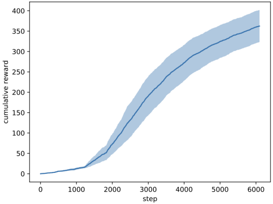
(b) Epsilon Greedy



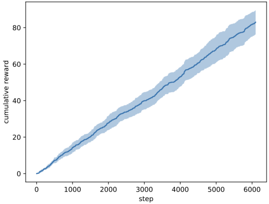
(c) Bootstrap-TS



(d) Rome-UCB



(e) Rome-TS



(f) Uniform Random

Conclusions & Future Work

- Introduced *residual overfit* and argued for its value in approximating model uncertainty.

Conclusions & Future Work

- Introduced *residual overfit* and argued for its value in approximating model uncertainty.
- Residual overfit method of exploration an "embarrassingly practical to implement" method for exploration (can be built in a day).

Conclusions & Future Work

- Introduced *residual overfit* and argued for its value in approximating model uncertainty.
- Residual overfit method of exploration an "embarrassingly practical to implement" method for exploration (can be built in a day).
- Further work on theory and to study which overfitting work best.

Conclusions & Future Work

- Introduced *residual overfit* and argued for its value in approximating model uncertainty.
- Residual overfit method of exploration an "embarrassingly practical to implement" method for exploration (can be built in a day).
- Further work on theory and to study which overfitting work best.
- Slides and further info at jamesmc.com/rome

Conclusions & Future Work

- Introduced *residual overfit* and argued for its value in approximating model uncertainty.
- Residual overfit method of exploration an "embarrassingly practical to implement" method for exploration (can be built in a day).
- Further work on theory and to study which overfitting work best.
- Slides and further info at jamesmc.com/rome

Thanks!