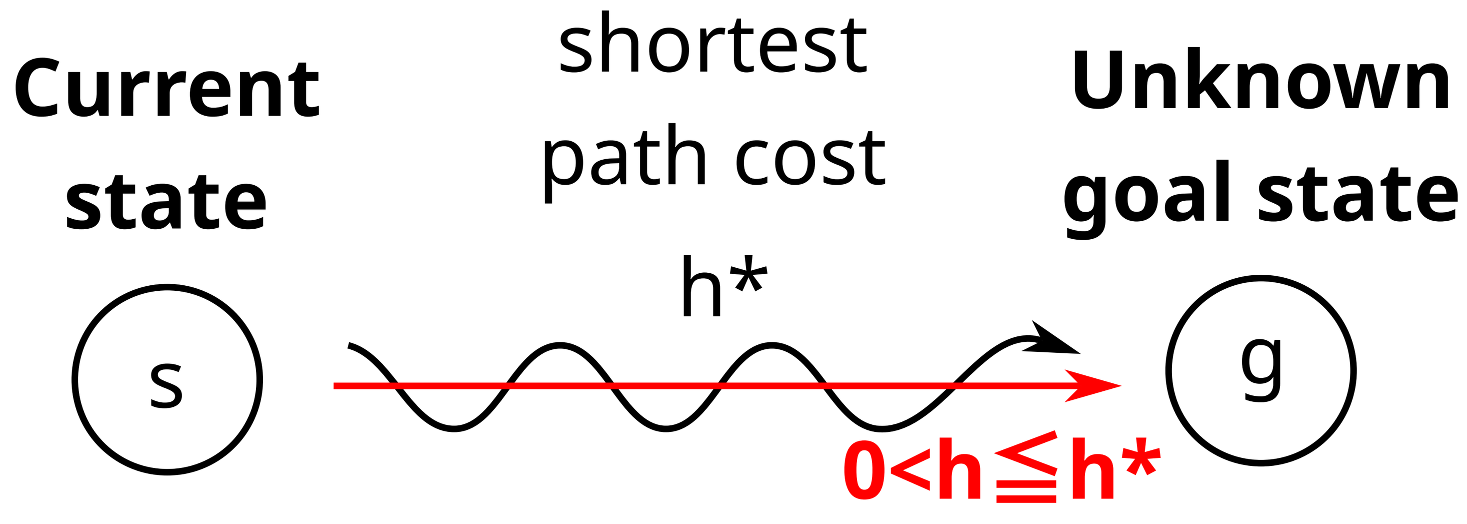
On Using Admissible Bounds for Learning Forward Search Heuristics

Carlos Núñez-Molina*, Masataro Asai*,
Pablo Mesejo, Juan Fernández-Olivares

MIT-IBM Watson AI Lab

1. Task: Learn the cost to go for forward heuristic search



We know a certain lower bound of h*:

Admissible heuristics h

obtained by solving a simplified problem

2. Stop using (Mean) Square Errors!

x : data point

μ : prediction (from **s, g**)

$(x-\mu)^2$: **Square error**

because it is just a special case of

Negative Log Likelihood (NLL)

of a Gaussian distribution

$$-\log p(x) = -\log \mathcal{N}(\mu, \sigma)$$

$$= \frac{(x-\mu)^2}{2\sigma^2} + \log\sqrt{2\pi\sigma^2}.$$

which contains the **square error**

It is a special case because

it sets an arbitrary fixed σ= $\frac{1}{\sqrt{2}}$

$$-\log \mathcal{N}(\mu, \frac{1}{\sqrt{2}}) = \frac{(x-\mu)^2}{2\cdot\frac{1}{2}} + \log\sqrt{2\pi\cdot\frac{1}{2}}$$

$$= (x-\mu)^2 + \text{Const.}$$

Choosing a loss function ⇔ Choosing a distribution

So don't "innovate" a new loss.
Innovate a distribution instead!

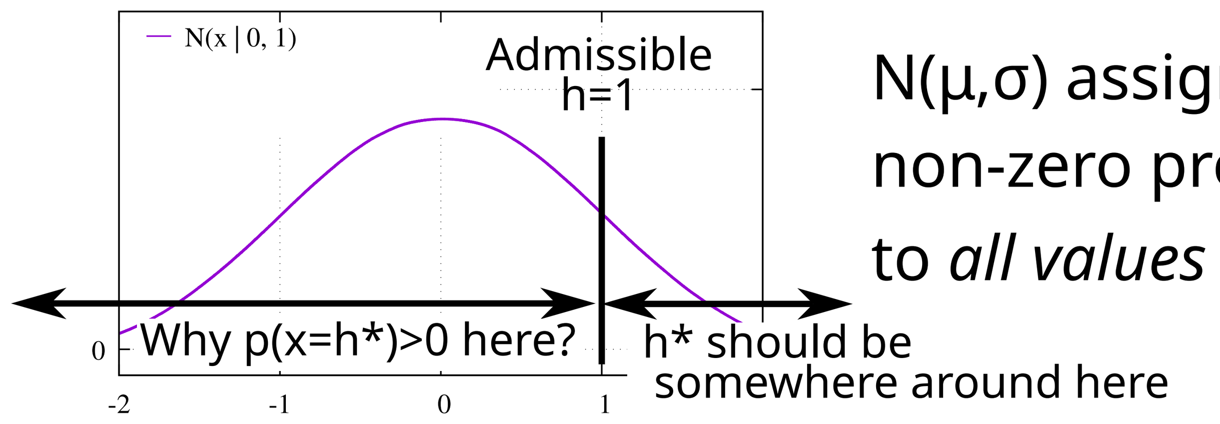
(Mean) Absolute error ⇔ Laplacian NLL

$$-\log \mathcal{L}(\mu, b) = \frac{|x-\mu|}{b} + \log 2b$$

(Mean) Squared error ⇔ Gaussian NLL

$$-\log \mathcal{N}(\mu, \sigma) = \frac{(x-\mu)^2}{2\sigma^2} + \log\sqrt{2\pi\sigma^2}$$

3. Thinking with Distributions: Is N(μ,σ) the correct distribution for h*?



N(μ,σ) assigns non-zero probability to *all values* x ∈ R

- We know **h* ≧ 0**
- We know **h* ≧ h : admissible heuristics**
- **Why** do we assign **non-zero probability** to h*<h?
- i.e. N(μ,σ) **ignores** our **expert knowledge** on h*
- It is **not** the **correct** distribution

4. What is *the correct* distribution?

↪Follow the Maximum Entropy Principle:

Jaynes 1957

*Choose the distribution with the **largest entropy** among those that satisfy the expert knowledge / constraint.*

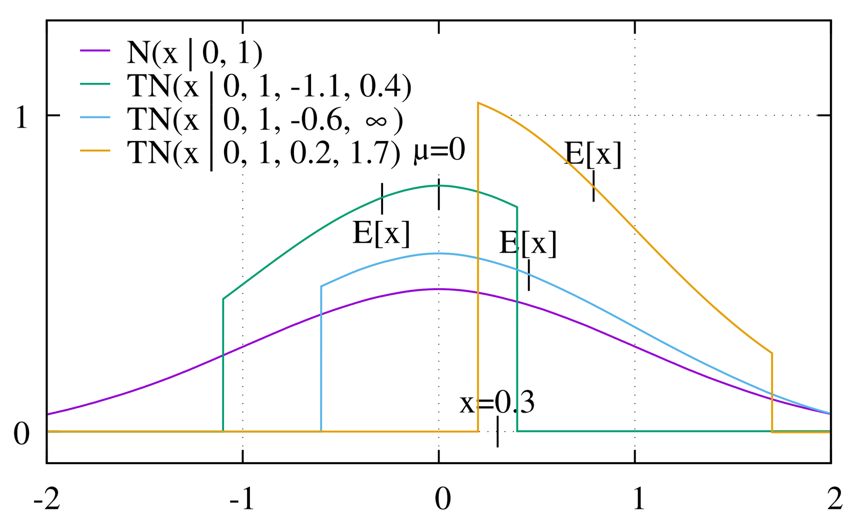
↪ We know a lower bound h ≦ h* !

Truncated-Gaussian distribution TN(μ, σ, l, u):

Dawson, Wragg 1973

Max-ent distribution under

- mean μ,
- variance σ^2,
- lower/upper bound l, u
  (u = ∞ is permissible)
  (l = LMcut heuristics)



NLL Loss:

$$-\log \mathcal{TN}(x|\mu,\sigma,l,u) = \frac{(x-\mu)^2}{2\sigma^2} + \log\sqrt{2\pi\sigma^2} + \log(\Phi(\tfrac{u-\mu}{\sigma}) - \Phi(\tfrac{l-\mu}{\sigma})).$$

l=LMcut is different across the dataset

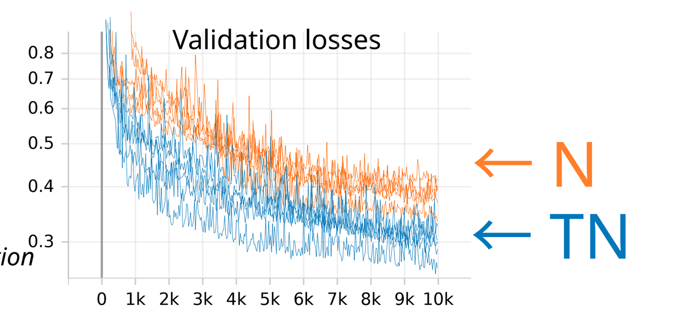
We use E[x] as the heuristics for GBFS. l ≦ E[x] ≦ u

given by a different formula (omitted due to space)

5. Results

Data generation details

- On 4 domains, we generated train, validation, test problems separately
- Train: 456-1536 instances, val/test: 132-384 instances
- *The test instances are larger than the training instances. → Evaluate problem-size generalization*
- h* : A*+LMcut on Fast Downward under 5min/8GB (train/val) or 30min/8GB (test)



TN training: faster & more accurate

h^{FF} vs. none: μ = NN output + h^{FF} vs. μ= NN output
learn vs. fixed: σ = NN output vs. σ= $\frac{1}{\sqrt{2}}$
+clip: Clip μ with lmcut when computing the MSE.

Training details:

- 40k steps
- AdamW batch-size 256 decay 10^{-2}
- gradient clip 0.1, lr=10^{-2}
- 0.5-2 hours / training

Findings:

- **N vs TN: TN outperforms N**
  - N often completely fails esp. w/o h^{ff}
- **Residual learning (h^{FF})** is effective
- **Learning σ** is crucial for TN
- **N+clip vs TN: Clipping** is not enough

domain	metric	h^{FF}	h^{LMcut}	learn/h^{FF} $\mathcal{N}$	learn/h^{FF} $\mathcal{TN}$	learn/none $\mathcal{N}$	learn/none $\mathcal{TN}$	fixed/h^{FF} $\mathcal{N}$	fixed/h^{FF} $\mathcal{TN}$	fixed/none $\mathcal{N}$	fixed/none $\mathcal{TN}$
blocks	MSE	22.8	25.06	.76±.1	**.65±.1**	3.26±.6	**2.71±.4**	.83±.1	**.66±.1**	2.97±.9	**2.44±.3**
	+clip			.76±.2		2.91±.4		.83±.2		2.74±.6	
ferry	MSE	9.77	11.10	3.73±.7	**3.45±.8**	141.05±29.4	**8.63±2.7**	**2.98±1.4**	3.85±.9	118.59±10.4	**9.58±1.5**
	+clip			3.72±.6		10.44±28.4		**2.98±1.1**		10.50±9.6	
gripper	MSE	9.93	15.82	3.65±.9	3.70±.9	68.12±16.0	**5.65±1.3**	3.69±.9	3.72±.9	68.22±16.1	**11.97±2.2**
	+clip			3.65±.7		13.37±15.2		3.69±.8		13.38±14.5	
visitall	MSE	13.9	36.4	7.67±.4	**5.30±.6**	25.31±7.9	**9.70±1.6**	6.49±.6	6.62±.9	21.71±2.6	**14.11±1.0**
	+clip			7.60±.4		18.79±7.3		**6.35±.6**		16.38±2.3	

Test metrics for NLM (smaller the better). Each number represents the mean±std of 5 random seeds. For each configuration, we performed 10^4 training steps, saving the checkpoints with the best validation MSE metric.

TN searches faster

- Instances: 100 instances subsampled from test
  ↪ **Larger than training**
- Pyperplan GBFS with early-stopping (added)
- Limited to 10k node evaluations
- Compared **success ratio** and **avg. evaluations**

Findings:

- **TN outperforms N, N+clip**
- **TN also outperforms h^{FF}**
- Better to **learn σ** and **learn the residual from h^{FF}**
  ↪ TN's heuristics E[x] depends on σ

domain	h^{FF}	learn/h^{FF} (proposed) $\mathcal{N}$	learn/h^{FF} (proposed) $\mathcal{N}$+clip	learn/h^{FF} (proposed) $\mathcal{TN}$	fixed/none (baseline) $\mathcal{N}$	fixed/none (baseline) $\mathcal{N}$+clip	fixed/none (baseline) $\mathcal{TN}$
Ratio of solved instances under 10^4 evaluations (higher the better)							
blocks	.13	.84±.19	.85±.19	**.88±.14**	**.79±.29**	.50±.35	.55±.33
ferry	.82	.91±.19	.91±.19	**.98±.05**	.01±.01	.57±.10	**.58±.13**
gripper	.96	**1**	**1**	**1**	0	.92±.12	**1**
visitall	.86	.97±.07	**.98±.06**	**.98±.05**	.82±.33	**1**	**1**
Average node evaluations (smaller the better)							
blocks	9309	2690±2128	2681±2121	**2060±1607**	**4118±2663**	6268±2675	5903±2685
ferry	5152	3216±1964	3117±1967	**2477±1093**	9933±92	6675±582	**6475±725**
gripper	3918	1642±139	1643±141	**1637±492**	10000±0	2941±1513	**1709±658**
visitall	3321	2156±1451	2148±1511	**1683±1290**	3384±3448	**591±216**	612±363

- Model-agnostic: tested NLM, GNN, linear regression
- Comparable performance to rank-based approach

notes:

- Chrestien et. al. uses HGN → we compared our HGN models vs Chrestien et. al.
- Our HGN models is not our best model (it performs worse than NLM models.)

domain	h^{FF}	learn/h^{FF} $\mathcal{N}$	learn/h^{FF} $\mathcal{N}$+clip	learn/h^{FF} $\mathcal{TN}$	(Chrestien *et al.*, 2023) L^{GBFS}
Ratio of solved instances under 10^4 evaluations (higher the better)					
blocks	.13	.70±.30	**.72±.29**	.48±.26	.24
ferry	**.82**	.01±.02	.01±.02	.02±.06	.61
gripper	**.96**	.36±.14	.37±.15	.39±.13	–
visitall	.86	**.99±.03**	.97±.04	.97±.04	0
Average node evaluations (smaller the better)					
blocks	9309	3984±2675	**3906±2649**	5844±2088	8282.6
ferry	5152	9916±132	9915±134	9834±424	**5103.5**
gripper	**3918**	7078±971	7008±1058	6949±1020	–
visitall	3321	1512±1192	1555±1149	**1472±1182**	10000