### Introduction

- Most models and the algorithms for fitting them have hyperparameters \( \eta \), e.g., number of layers in a neural network, gradient descent learning rate parameters, number of topics in a topic model, prior variance.
- Existing methods for choosing them include expert knowledge, grid search, random sampling, or Bayesian optimization (BayesOpt) [Snoek et al. 2012].
- BayesOpt is an automated way of choosing hyperparameters without doing an exhaustive grid search. In more detail:
  - It posits a probabilistic performance model \( \mathcal{P} \) that captures how well the target model performs as a function of \( \eta \).
  - It uses the performance model to find where to explore next in hyperparameter space so as to find the best performing \( \eta \).
- BayesOpt is therefore an empirical Bayes method for finding the optimal hyperparameter \( \tilde{\eta} = \arg \max_{\eta} \mathbb{E}_{\mathcal{P}(\eta | \theta)}[p(X | \theta, \eta)] \).
- Is optimization the best we can do for hyperparameter tuning?

### The Case for Hyperparameter Averaging

- BayesOpt requires finding a set of expensive intermediate posteriors \( \{p(\theta | X, \eta^{(1)}), \ldots, p(\theta | X, \eta^{(S)})\} \) but then throws most of these away and only uses the best one \( p(\theta | X, \tilde{\eta}) \).
- In contrast, hyperparameter averaging makes use of the intermediate posteriors and takes account the fact that we always harbour uncertainty about the choice of \( \eta \) on finite validation sets.
- For example, consider maximizing marginal likelihood w.r.t. number of topics in latent Dirichlet allocation (LDA), see right. The optimal choice on a held out validation set is not optimal on test data.

### Empirical Bayes for Hyperparameter Averaging (EB-Hyp)

- The starting point for EB-Hyp is Bayes empirical Bayes [Carlin & Louis, 2000] which maximizes marginal likelihood w.r.t. hyperparameter \( \lambda \):
  \[
  \hat{\lambda} = \arg \max_\lambda \mathbb{E}_{\mathcal{P}(\lambda | \theta, \eta)}[p(X | \theta, \eta)],
  \]
  for some choice of hyperprior \( p(\eta | \lambda) \).
- Procedure repeats for iterations \( s = 1, \ldots, S \):
  1. draw \( \tilde{\lambda}^{(s)}(X | \eta) \sim p(X | \eta) \mathcal{P}(\tilde{\lambda}^{(s)} | \eta) \) from \( \eta^{(s)} \)
  2. draw \( \eta^{(s)} \sim \mathcal{P}(\eta | X) \)
  3. evaluate \( f_{X|\tilde{\lambda}^{(s)}} = \int p(X | \theta) p(\theta | \eta^{(s)}) d\theta \)
- Natural choice for \( \mathcal{P} \) is Gaussian process (GP) for its convenient marginalization properties.

### The Train-Marginalize-Test Pipeline

- EB-Hyp avoids overfitting training data through marginalization and allows us to train, marginalize, and test without a separate validation data set.
- It consists of three steps:
  - **Train** a set of parameters on training data \( X \), each one conditioned on a choice of hyperparameter \( \eta \).
  - **Marginalize** \( \eta \) out of the set of full or approximate posteriors.
  - **Test** (or Deploy) the marginal predictive distribution on test data \( X_{\text{test}} \) and report the performance.
- This framework simplifies the evaluation and deployment pipeline.

### EB-Hyp Algorithm

**Algorithm 1** Pseudocode for EB-Hyp

1. inputs training data \( X \), inference algorithm \( \mathcal{A} : (X, \eta) \rightarrow p(\theta | X, \eta) \), set of posteriors \( p(\theta | \text{train}, \eta^{(s)}) \)
2. output predictive density \( p(X_{\text{test}} | X) \), optimal hyper-hyperparameter \( \hat{\lambda} \)
3. while \( V \) not converged do
4. draw performance \( f^* \) from training GP \( \tilde{\lambda}^{(s)}(X | \eta) \sim \mathcal{GP}(\cdot | \mathcal{P}, \mathcal{V}) \)
5. calculate hyperparameter posterior \( \tilde{\eta}^{(s)}(X) \leftarrow Z^{-1} \tilde{\lambda}^{(s)}(X | \eta)p(\eta) \)
6. draw next evaluation point \( \eta^{(s)} \leftarrow \arg \max_\eta \tilde{\lambda}^{(s)}(X | \eta) \)
7. run parameter inference conditioned on \( \eta^{(s)} \) w.r.t.
8. evaluate performance \( f^{(s)}_{X|\tilde{\lambda}^{(s)}} \leftarrow \int p(X | \theta)p(\theta | \eta^{(s)})d\theta 
9. append \( (\eta^{(s)}, f^{(s)}_{X|\tilde{\lambda}^{(s)}}) \) to history \( V \)
10. end while
11. find optimal \( \lambda \).
12. return approximation to \( p(X_{\text{test}} | X) \) and \( \lambda \).

### Results

- Predictive log lik. for deep latent Gaussian model, Labeled Faces in the Wild
  - Method: EB-Hyp with validation
  - Predictive Log Lik.: -357650 (-0.00%)
  - (% Improvement on BayesOpt: -361661 (-1.12%)
  - Random: -366074 (-64.5%)

- Predictive log lik. for LDA, 20 Newsgroup dataset
  - Method: EB-Hyp with validation
  - Predictive Log Lik.: -375488 (-0.00%)
  - (% Improvement on BayesOpt: -361661 (-1.12%)
  - Random: -366074 (-64.5%)

### Conclusions

- Introduced a general-purpose procedure for dealing with unknown hyperparameters that control the behaviour of machine learning models.
- EB-Hyp is based on approximately marginalizing the hyperparameters by taking a weighted average of posteriors calculated by existing inference algorithms that are time intensive.
- This work points toward a tendency of the standard optimization-based methods to overfit hyperparameters. Other things being equal, this tendency punishes methods that are more sensitive to hyperparameters compared to methods that are less sensitive. The result is a bias in the literature towards methods whose generalization performance is less sensitive to hyperparameters. Averaging approaches like EB-Hyp help reduce this bias.

### Table of Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \theta )</td>
<td>the model parameters</td>
</tr>
<tr>
<td>( \eta )</td>
<td>the hyperparameters</td>
</tr>
<tr>
<td>( \lambda )</td>
<td>the hyper-hyperparameters</td>
</tr>
<tr>
<td>( \tilde{\eta} )</td>
<td>the hyperparameters fit by empirical Bayes</td>
</tr>
<tr>
<td>( \hat{\lambda} )</td>
<td>the hyper-hyperparameters fit by empirical Bayes</td>
</tr>
<tr>
<td>( X )</td>
<td>training data</td>
</tr>
<tr>
<td>( X_{\text{test}} )</td>
<td>unseen test data</td>
</tr>
</tbody>
</table>

### References


---

http://jamesmc.com/eb-hyp.html  
james@spotify.com