HT2015: SC4
Statistical Data Mining and Machine Learning

Dino Sejdinovic
Department of Statistics
Oxford

http://www.stats.ox.ac.uk/~sejdinov/sdmml.html

# Model Complexity and Generalization

Generalization

- Generalization ability: what is the out-of-sample error of learner f?
- training error $\neq$ testing error.
- We learn f by minimizing some variant of empirical risk $R^{\text{emp}}(f)$- what can we say about the true risk $R(f)$?
- Two important factors determining generalization ability:
    - Model complexity
    - Training data size

Learning Curves
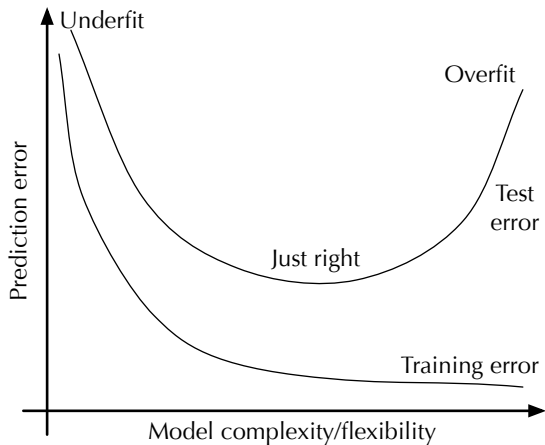
Fixed dataset size, varying model complexity.

Learning Curves
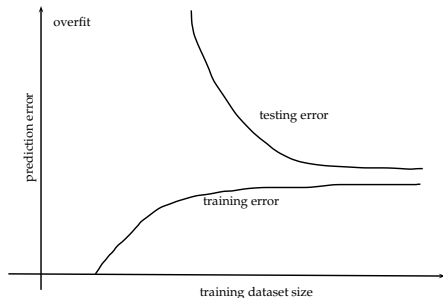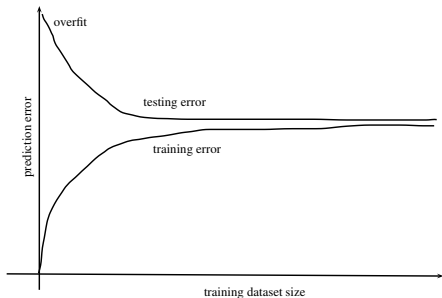
Fixed model complexity, varying dataset size.
Two models: one simple, one complex. Which is which?

Bias-Variance Tradeoff

- Where does the prediction error come from?
- Example: Squared loss in regression: $\mathcal{X} = \mathbb{R}^p$, $\mathcal{Y} = \mathbb{R}$,

$$L(Y, f(X)) = (Y - f(X))^2$$

- Optimal f is the conditional mean:

$$f_*(x) = \mathbb{E}[Y | X = x]$$

- Follows from $R(f) = \mathbb{E}_X \mathbb{E}\left[(Y - f(X))^2 \Big| X\right]$ and

$$
\begin{aligned}
&\mathbb{E}\left[(Y - f(X))^2 \Big| X = x\right] \\
=\ &\mathbb{E}\left[Y^2 \Big| X = x\right] - 2f(x)\,\mathbb{E}[Y | X = x] + f(x)^2 \\
=\ &\mathsf{Var}[Y | X = x] + (\mathbb{E}[Y | X = x] - f(x))^2.
\end{aligned}
$$

Bias-Variance Tradeoff

- Optimal risk is the intrinsic conditional variability of Y (noise):

$$R(f_*) = \mathbb{E}_X \left[\text{Var} \left[Y | X \right] \right]$$

- **Excess risk**:

$$R(f) - R(f_*) = \mathbb{E}_X \left[\left(f(X) - f_*(X) \right)^2 \right]$$

- How does the excess risk behave **on average**?
- Consider a randomly selected dataset $\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^n$ and $f^{(\mathcal{D})}$ trained on $\mathcal{D}$ using a model (hypothesis class) $\mathcal{H}$.

$$
\begin{aligned}
\mathbb{E}_{\mathcal{D}} \left[R(f^{(\mathcal{D})}) - R(f_*) \right] &= \mathbb{E}_{\mathcal{D}} \mathbb{E}_X \left[\left(f^{(\mathcal{D})}(X) - f_*(X) \right)^2 \right] \\
&= \mathbb{E}_X \mathbb{E}_{\mathcal{D}} \left[\left(f^{(\mathcal{D})}(X) - f_*(X) \right)^2 \right].
\end{aligned}
$$

Bias-Variance Tradeoff

- Denote $\bar{f}(x) = \mathbb{E}_{\mathcal{D}} f^{(\mathcal{D})}(x)$ (average decision function over all possible datasets)

$$\mathbb{E}_{\mathcal{D}} \left[\left(f^{(\mathcal{D})}(X) - f_*(X) \right)^2 \right] = \underbrace{\mathbb{E}_{\mathcal{D}} \left[\left(f^{(\mathcal{D})}(X) - \bar{f}(X) \right)^2 \right]}_{\text{Var}_X(\mathcal{H}, n)} + \underbrace{\left(\bar{f}(X) - f_*(X) \right)^2}_{\text{Bias}_X^2(\mathcal{H}, n)}$$
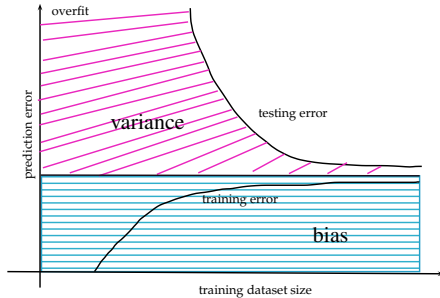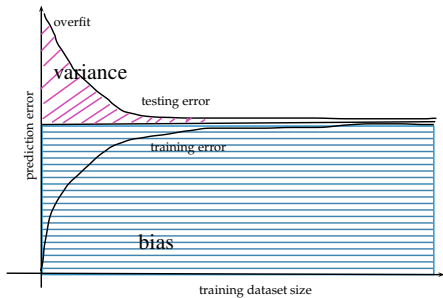
Now,

$$\mathbb{E}_{\mathcal{D}} R(f^{(\mathcal{D})}) = R(f_*) + \mathbb{E}_X \text{Var}_X(\mathcal{H}, n) + \mathbb{E}_X \text{Bias}_X^2(\mathcal{H}, n)$$
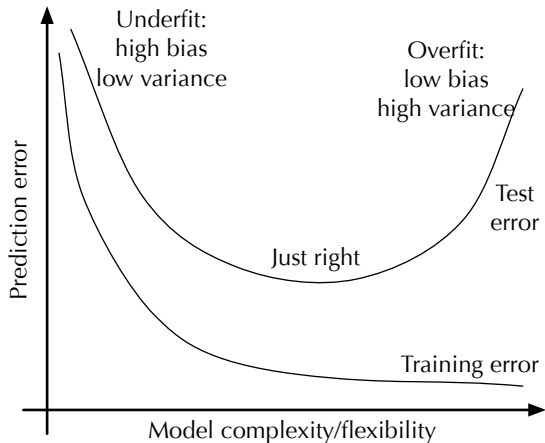
Where does the prediction error come from?

- **Noise**: Intrinsic difficulty of regression problem.
- **Bias**: How far away is the best learner in the model (average learner over all possible datasets) from the optimal one?
- **Variance**: How variable is our learning method if given different datasets?

Learning Curves

Learning Curve

Validation and Cross-Validation

Empirical vs True Risk

- In general,
$$R(f) = R^{\text{emp}}(f) + \text{overfit penalty.}$$

- Overfit penalty depends on the complexity of the model (**VC analysis**).
- Regularization: **approximate the overfit penalty**. More complex the model, larger the overfit penalty.
- (Cross-)Validation: try to **estimate** $R(f)$ **directly**.
- For any example not used in training:
$$\mathbb{E}\left[L\left(y_{\text{test}}, f(x_{\text{test}})\right)\right] = R(f).$$

- But for examples used in training:
$$\mathbb{E}\left[L\left(y_{\text{train}}, f(x_{\text{train}})\right)\right] \neq R(f).$$

Optimizing Tuning Parameters

- How can we optimize generalization ability, via optimizing choice of tuning parameters, model size, and learning parameters?

- Suppose we have split data into training/test set.

- Test set can be used to determine generalization ability, and used to choose best setting of tuning parameters/model size/learning parameters with best generalization.

- Once these tuning parameters are chosen, still important to determine generalization ability, but cannot use performance on test set to gauge this anymore!

- Idea: split data into 3 sets: training set, test set, and **validation set**.
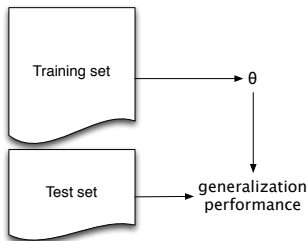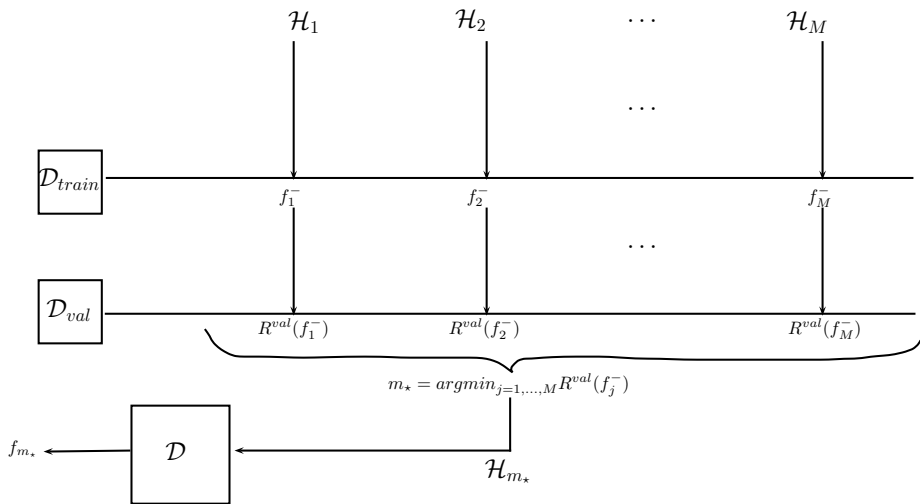
Validation error

- **Out-of-sample average loss**. For a dataset $\{\tilde{x}_i, \tilde{y}_i\}_{i=1}^{v}$ unseen in training

$$R^{\text{val}}(f) = \frac{1}{v} \sum_{i=1}^{v} L\left(\tilde{y}_i, f(\tilde{x}_i)\right)$$

- $\mathbb{E}\left[R^{\text{val}}(f)\right] = R(f)$, $\text{Var}\left[R^{\text{val}}(f)\right] \asymp \frac{1}{v}$, i.e. $R^{\text{val}}(f) = R(f) \pm \mathcal{O}\left(\frac{1}{\sqrt{v}}\right)$
- Just like testing error so far.
- It becomes validation error only once it is used to **change our learning**.

Validation

Validation

- For each combination of tuning parameters γ:
  - Train our model on the training set, fit parameters $\theta = \theta(\gamma)$, obtaining decision function $f_{\theta(\gamma)}$.
  - Evaluate $R^{\text{val}}\left(f_{\theta(\gamma)}\right)$ average loss on a validation set.
- Pick γ^* with best performance on validation set.
- Using γ^*, train on both training and validation set to obtain the optimal θ^*.
- $R^{\text{val}}(f_{\theta(\gamma^*)})$ is now a biased estimate of $R(f_{\theta(\gamma^*)})$ and can be overly optimistic!
- Evaluate model with γ^*, θ^* on test set, reporting generalization performance.
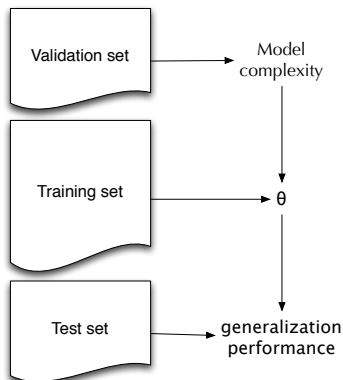
Bias introduced by validation

- **Example**: Selecting between two equally bad classifiers f_1 and f_2:

$$R(f_1) = R(f_2) = 0.5.$$

- Assume that we have independent unbiased estimators $R_1 = R^{\mathrm{val}}(f_1)$, $R_2 = R^{\mathrm{val}}(f_2)$, both uniform on $[0, 1]$
- Learning rule $f_\star$ chosen to minimize R^{val} is either f_1 or f_2, so also equally bad.
- But $\mathbb{E}\min\{R_1, R_2\} = \frac{1}{3}$, so in terms of validation error it may appear that we are getting an improvement!

Validation error and Generalization

How contaminated are different parts of data in terms of being able to tell us something about generalization ability?

- Training data: fully contaminated, used in learning - $R^{\text{emp}}(f)$ is usually far from $R(f)$ (unless the model is too simple for the amount of data).
- Validation data: partly contaminated, used in model selection / meta-learning - $R^{\text{val}}(f)$ is biased, but still useful, provided that:
    - we have a large enough validation set size v
    - we do not use it to select from a large number M of models
- Test data: clean, not used for any part of learning.

Typically,

$$R(f) \leq R^{\text{val}}(f) + \underbrace{\mathcal{O}\left(\sqrt{\frac{\log M}{v}}\right)}_{\text{overfit penalty of the meta-model}}$$
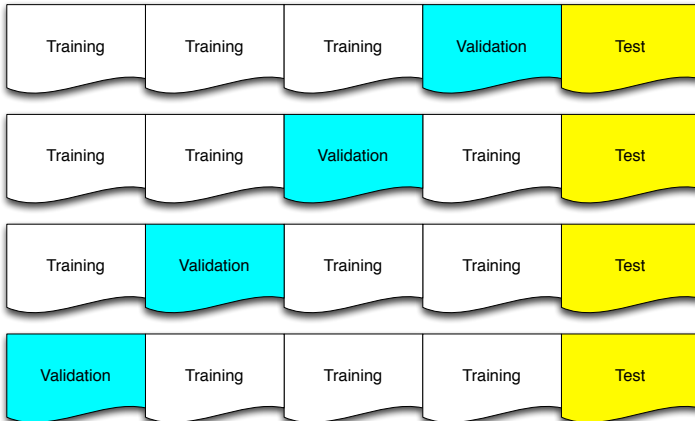
Size of validation set?

- In practice, there is just one dataset! If v is used for computing validation error, then only $n - v$ used for training.
  - Small v : $R^{\text{val}}(f^-)$ is a bad estimate of $R(f^-)$
  - Large v : $R^{\text{val}}(f^-)$ is a reliable estimate of a much worse risk (f^- learned on much less data than f)!
- We are using:

$$R(f) \underset{(\text{need small } v)}{\approx} R(f^-) \underset{(\text{need large } v)}{\approx} R^{\text{val}}(f^-)$$

- Wasteful to split into 3 subsets.
- Different approach: **cross-validation**.
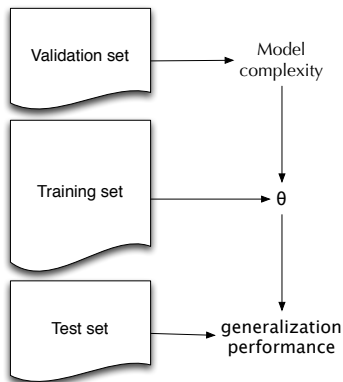
Cross-Validation

Cross-Validation

- Basic approach:
    - Split training set into T folds.
    - For each γ and each $t = 1, \ldots, T$:
        - Use fold t as validation set and the rest to train the model parameters $\theta_t = \theta_t(\gamma)$, obtaining decision function $f_{t,\gamma}^-$.

    $$R_t^{\text{val}}(f_{t,\gamma}^-) = \frac{1}{|\text{Fold}(t)|} \sum_{i \in \text{Fold}(t)} L(y_i, f_{t,\gamma}^-(x_i))$$

    - Choose γ^* which minimizes validation error averaged over folds

    $$\frac{1}{T} \sum_{t=1}^{T} R_t^{\text{val}}(f_{t,\gamma}^-)$$

    - Train model with tuning parameter γ^* on all training set to obtain f_{γ^*}.
    - Report generalization performance on test set.
- **Leave-One-Out (LOO)** cross validation: one data item per fold, i.e., $T = n$.

Cross-validation can be computationally expensive ($T\times$ increase in complexity).

Leave-One-Out Cross-Validation

Leave-one-out (LOO) cross validation: one data item per fold, i.e., $T = n$.

- Since only one data item not used in training, $R(f_{t,\gamma}^-)$ are all very close to $R(f_\gamma)$ (small v benefit).
- Thus,

$$\frac{1}{n}\sum_{t=1}^{n} R_t^{\mathsf{val}}(f_{t,\gamma}^-) = \frac{1}{n}\sum_{t=1}^{n} L(y_t, f_{t,\gamma}^-(x_t))$$

  has a small variance (large v benefit).

- All examples for validation and all examples for training.
- summands are no longer independent