Statistical Learning and Applications

Laurent Jacob

October 2, 2014

Summary of the previous class

- Penalized risk minimization
- Ridge penalty, ridge regression, SVM.
- Fundamentals of constrained optimization.

You now know one regression algorithm, one classification algorithm and why they make sense.

Outline for supervised learning

1. Support vector machines (continued).
2. ℓ_1 penalties.
3. Cross validation.
4. Local methods (nearest neighbors, smoothing).

Support vector machines

Summary

- Linear method for binary classification.
- Hard margin: for linearly separable problems, find the separating hyperplane with largest margin.
- Soft margin: allow points to be on the wrong side of the margin, but charge it to the objective function.
- Soft margin can be written equivalently as a penalized empirical risk minimization problem (hinge loss, ridge penalty).
- Primal/Dual problems for both methods.

Remarks

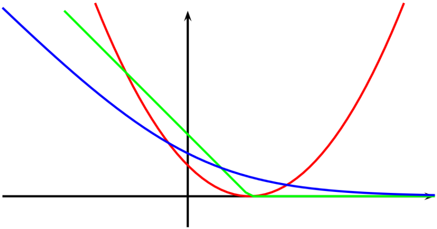
- We started from a different perspective (maximize margin) and showed retrospectively that the problem we solved could be thought of as penalized empirical risk minimization.
- Yields another interpretation for ℓ_2 regularization of linear functions.
- In practice controlling this trade-off makes sense even if the classes are linearly separable (as discussed during the first class).

Logistic regression (1/2)

A similar analysis can be made for logistic regression:

- Can be derived from a Bernouilli model: $\mathbf{E}\left[y_i|x_i\right] = p_i = \frac{1}{1+e^{-w^\top x_i}}$, where the logistic function ensures that $p_i \in [0,1]$.
- Leads to a linear separation: $\ln\left(\frac{p_i}{1-p_i}\right) = w^\top x_i$.
- Maximizing the negative log likelihood yields $\min_w \sum_{i=1}^n \ln(1+e^{-w^\top x_i})$.

Logistic regression (2/2)



- Empirical risk for a loss function with very similar shape (and behavior) as the hinge loss.
- Intuition/justification is important but can be deceiving. In the end, it is crucial to compare objectives.

Algorithms

We mentioned hard and soft margin SVM could be written as a QP with box constraints. In practice however, faster dedicated algorithms were proposed, *e.g.*,

SimpleSVM

- Active set method: solve sub-problem with a restricted set of points, iteratively add the ones which most violate the constraints.
- Efficient when only a few α_i are non-zero (small C).

Stochastic gradient descent

- Take gradient steps with respect to randomly drawn single points.
- Efficient when the number of samples is large ("Large scale learning").

ℓ_1 penalty

Penalties: the ℓ_1 norm

- Another popular penalty is the ℓ_1 norm:

$$\Omega(\theta) = \|\theta\|_1 \overset{\Delta}{=} \sum_{j=1}^{p} |\theta_j|.$$

- Interesting property: leads in practice to estimators

$$\hat{\theta} \in \arg\min_{\theta} \sum_{i=1}^{n} L(y_i, \theta^\top x_i) + \lambda\|\theta\|_1$$

  which are **sparse**, *i.e.*, contain few non-zero values.

- Combined with the ℓ_2 loss: **Lasso** (Tibshirani et al., 1996) or **basis pursuit** (Chen et al., 1999).

Penalties: the ℓ_1 norm

- We can think of the problem constrained by the ℓ_1 norm as a **convex relaxation** of the one constrained by the ℓ_0 pseudo-norm:

$$\|\theta\|_0 \triangleq \sum_{j=1}^{p} \mathbf{1}_{\{\theta_j \neq 0\}}, \qquad \|\theta\|_1 \triangleq \sum_{j=1}^{p} |\theta_j|.$$

- More precisely, the convex envelope of a function f over a space $\mathcal{X}$ is the largest convex function underestimating f over $\mathcal{X}$.
- $\|.\|_1$ is the convex envelope of $\|.\|_0$ over $[-1, 1]^p$.
- **[Exercise] :** why $[-1, 1]^p$ and not $\mathbb{R}^p$?

Penalties: the ℓ_1 norm

- Contrarily to the ℓ_0 pseudo-norm, the ℓ_1 norm penalizes the amplitude of the θ_j, not only the fact that they are non-zero.
- Consequence of this double effect of the penalty: setting a lot of coefficients to 0 may shrink non-zero coefficients a lot.
- Frequent (heuristic) strategy:
    1. Use the ℓ_1 penalized estimator to get a sparse solution.
    2. Perform unpenalized estimation on the support of this function.

Sparsity of ℓ_1 penalized estimators

Analytical intuition (penalized form, special case)

$$\hat{\theta} \in \arg\min_{\theta} \frac{1}{2}(y-\theta)^2 + \lambda|\theta|$$

- The optimum is characterized by the stationarity condition:

$$\partial_\theta \left(\frac{1}{2}(y-\theta)^2 + \lambda|\theta| \right) \ni 0,$$

where ∂_θ is the **subdifferential** operator.

Sparsity of ℓ_1 penalized estimators: subdifferential

- The absolute value is not differentiable at 0.
- Like for any convex function, we can however define its subdifferential:

$$\partial_\theta f(\theta_0) \overset{\Delta}{=} \left\{ s : f(\theta) \geq f(\theta_0) + s(\theta - \theta_0) \right\} .$$

- **[Exercise :]** Compute the subdifferential of the absolute value at 0.

Sparsity of ℓ_1 penalized estimators

Analytical intuition (penalized form, special case)

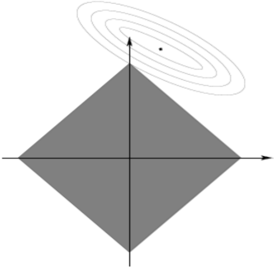
$$\partial_\theta \left(\frac{1}{2}(y-\theta)^2 + \lambda|\theta| \right) \ni 0,$$

- The stationarity condition is that either
  - $\theta = 0$ and $(y-\theta) \in [-\lambda, \lambda]$, or
  - $|\theta| > 0$ and $\theta - y + \lambda \mathrm{sign}(\theta) = 0$.
- We can therefore check that the solution is $\hat{\theta} = \mathrm{sign}(y)(|y| - \lambda)^+$, which involves the **soft thresholding** operator $(.)^+ \triangleq max(., 0)$.
- If $|y| < \lambda$, the estimator is set to 0. Otherwise, its amplitude is decreased by λ.

Sparsity of ℓ_1 penalized estimators

Geometrical intuition (constrained form)

$$\hat{\theta} \in \arg\min_{\|\theta\|_1 \leq \mu} \sum_{i=1}^{n} L(y_i, \theta^\top x_i)$$



The constrained minimum is likely to lie on one of the singularities.

Sparsity of ℓ_1 penalized estimators

Geometrical intuition (constrained form, special case)

$$\hat{\alpha} \in \underset{\|\alpha\|_1 \leq \mu}{\arg\min} \frac{1}{2}\|\alpha - x\|^2$$

- Equivalent problem with indicator function: $\min_\alpha \frac{1}{2}\|x - \alpha\|^2 + \delta_{\|\alpha\|_1 \leq \mu}(\alpha)$.
- Optimality given by stationarity condition (unconstrained convex problem):

  $$\partial_\alpha \left(\frac{1}{2}\|\alpha - x\|^2 + \delta_{\|\alpha\|_1 \leq \mu}(\alpha) \right) \ni 0,$$

  where ∂_α now denotes the **subgradient** of the (convex but non differentiable) objective.

Sparsity of ℓ_1 penalized estimators

Exercise:

- Show that the subgradient of $\delta_{\|\alpha\|_1 \leq \mu}$ at a point $\bar{\alpha} \leq \mu$ is the normal cone to the ball at $\bar{\alpha}$:

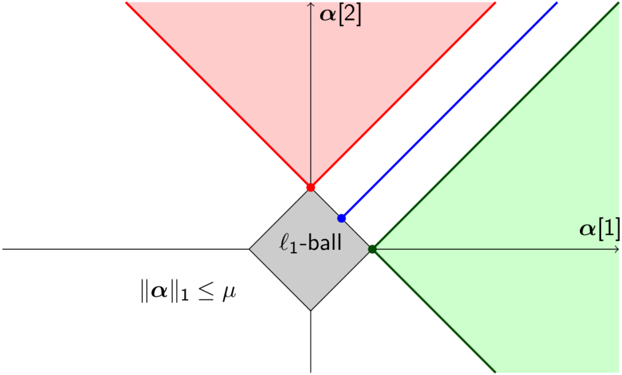
$$N(\bar{\alpha}) = \left\{ d \in \mathbb{R}^p : d^\top (\alpha - \bar{\alpha}) \leq 0 \quad \forall \|\alpha\|_1 \leq \mu \right\}.$$

- Describe the set of points in $\mathbb{R}^p$ which project at a given point $\bar{\alpha}$ of the ball.

Sparsity of ℓ_1 penalized estimators

Geometrical intuition (constrained form, special case)

$$\hat{\alpha} \in \underset{\|\alpha\|_1 \leq \mu}{\arg\min} \|\alpha - x\|^2$$



Picture by J. Mairal

Sparsity of ℓ_1 penalized estimators

- We provided intuitive explanations as to why the ℓ_1 penalty led to sparse estimators.
- No theoretical result guarantees a fixed level of sparsity for a particular penalty intensity.
- However, we have a good empirical knowledge of how these estimators behave (very hot topic), and we know that in practice the ℓ_1 penalty allows to adjust the sparsity level.

Algorithms for the Lasso

$$\min_{\alpha} \|X\alpha - y\|^2 + \lambda\|\alpha\|_1.$$

- Can be formulated as a QP:

$$\min_{\alpha_-,\alpha_+\geq 0} \|X\alpha_+ - X\alpha_- - y\|^2 + \lambda\alpha_+^\top\mathbf{1} + \lambda\alpha_-^\top\mathbf{1}.$$

- Like for SVM, we can therefore use generic toolboxes.
- Much faster dedicated algorithms have been devised. Choice of best algorithm depends on exact setting.
- We detail one easy to implement and generally efficient method: coordinate descent.

Coordinate descent for the Lasso

- Iteratively fix all but one variables, optimize with respect to this variable:

$$\arg\min_{\alpha_j} \|X\alpha - y\|^2 + \lambda\|\alpha\|_1 = \arg\min_{\alpha_j} \|X_j\alpha_j - (y - X_{-j}\alpha_{-j})\|^2 + \lambda|\alpha_j|.$$

- Assuming the columns of X have unit ℓ_2 norm,

$$\alpha_j^* = \mathrm{sign}(X_j^\top r)\left[|X_j^\top r| - \lambda\right]^+,$$

  where $r = (y - X_{-j}\alpha_{-j})$ are the current residuals.
- Coordinate descent does not converge in general for non-smooth objectives. Was proved to converge in this case.
- Can be complemented with an **active set** strategy.

Penalties: the ℓ_1 norm

- Informally, we recognize the bias-variance tradeoff again. The ℓ_1 term biases the estimator towards 0 in a different way than the ridge.
- Formal analysis is harder than for the ridge because even with an ℓ_2 loss, there is no closed form for the estimator. Analysis uses optimality conditions.

Consistencies for the ℓ_1 norm

Assuming a linear model $y = x^\top \theta + \varepsilon$, three important questions are:

1. Consistency: does $\|\hat{\theta} - \theta\|$ converge to 0 as n tends to ∞?

2. Model selection consistency (sparsistency): does the zero pattern of $\hat{\theta}$ converge to the zero pattern of θ?

3. Prediction: does $\frac{1}{n}\|X\hat{\theta} - X\theta\|$ converge to 0 as n tends to ∞?

Consistencies for the ℓ_1 norm

Assuming a linear model $y = x^\top \theta + \varepsilon$, three important questions are:

1. Consistency: does $\|\hat{\theta} - \theta\|$ converge to 0 as n tends to ∞?
   Can be proved under **restricted eigenvalues** condition on design restricted to support of θ. Related to identifiability of θ (curvature of risk large enough in enough directions).
2. Model selection consistency (sparsistency): does the zero pattern of $\hat{\theta}$ converge to the zero pattern of θ?
3. Prediction: does $\frac{1}{n}\|X\hat{\theta} - X\theta\|$ converge to 0 as n tends to ∞?

Consistencies for the ℓ_1 norm

Assuming a linear model $y = x^\top \theta + \varepsilon$, three important questions are:

1. Consistency: does $\|\hat{\theta} - \theta\|$ converge to 0 as n tends to ∞?
   Can be proved under **restricted eigenvalues** condition on design restricted to support of θ. Related to identifiability of θ (curvature of risk large enough in enough directions).
2. Model selection consistency (sparsistency): does the zero pattern of $\hat{\theta}$ converge to the zero pattern of θ?
   Can be proved under identifiability over support of θ and **irrepresentable conditions**:

$$\max_{j \notin J} |X_j^\top X_J (X_J^\top X_J)^{-1}| \leq 1 - \gamma, \quad \gamma \in]0, 1],$$

   where J is the support of θ. Unrealistic for many problems (molecular data, images...).
3. Prediction: does $\frac{1}{n}\|X\hat{\theta} - X\theta\|$ converge to 0 as n tends to ∞?

Consistencies for the ℓ_1 norm

Assuming a linear model $y = x^\top \theta + \varepsilon$, three important questions are:

1. Consistency: does $\|\hat{\theta} - \theta\|$ converge to 0 as n tends to ∞?
   Can be proved under **restricted eigenvalues** condition on design restricted to support of θ. Related to identifiability of θ (curvature of risk large enough in enough directions).
2. Model selection consistency (sparsistency): does the zero pattern of $\hat{\theta}$ converge to the zero pattern of θ?
   Can be proved under identifiability over support of θ and **irrepresentable conditions**:

   $$\max_{j \notin J} |X_j^\top X_J (X_J^\top X_J)^{-1}| \leq 1 - \gamma, \quad \gamma \in]0, 1],$$

   where J is the support of θ. Unrealistic for many problems (molecular data, images...).
3. Prediction: does $\frac{1}{n}\|X\hat{\theta} - X\theta\|$ converge to 0 as n tends to ∞?
   Can be proved under rather general conditions.

Penalties: the ℓ_1 norm

- Sparsity is a desirable property because it leads to interpretable estimates.
- However it is important to be clear about the objective: are we trying to detect variables x associated with y (hypothesis testing), or to minimize the risk incurred when predicting y from $\theta^\top x$ (estimation/prediction)?
- Note that testing procedures can be derived — among other approaches — using a ridge or an ℓ_1 penalized estimator (much more complicated for the latter).

Penalties: variations

- Graph Laplacian,
- Trace norm,
- Fused norm,
- Group lasso,
- Weighted ℓ_1,
- Other ℓ_p norms,
- Group fused,
- Overlapping groups,
- All size k groups,
- Groups defined over a graph,
- Combinations
- ...

× combinations with various loss functions.

Relationship to maximum likelihood estimation

Relationship to maximum likelihood estimation

- Until now we discussed risk minimization without involving **models** for the data.
- This discussion and the related penalized methods are however related to methods based on the **likelihood** of data under some model.
- Given a model $p(D|\theta)$ of data D, for example

$$y = \bar{\theta}^\top x + \varepsilon,$$

where ε is endowed with some distribution, it is common to estimate θ by the value which maximizes the likelihood of the data under the model:

$$\hat{\theta} = \arg\max_{\theta} p(D|\theta).$$

(popularized by R. A. Fisher at the beginning of the 20th century).

Relationship to maximum likelihood estimation

- The maximum likelihood estimator has a few desirable asymptotic properties under some regularity conditions:
  - Consistency : $\hat{\theta}_{MLE} \rightarrow \bar{\theta}$ as n increases,
  - Asymptotic normality,
  - Efficiency (asymptotic minimal variance).
- Can be biased though.
- Can behave poorly when p/n is not small enough.

Relationship to maximum likelihood estimation

- In practice, it is often easier to minimize the **negative log likelihood**:

$$\hat{\theta} = \arg\min_{\theta} - \log p(D|\theta).$$

  We recover an empirical risk minimization problem, where the loss function is defined by $L(D, \theta) \overset{\Delta}{=} - \log p(D|\theta)$.

- **Exercise :** which loss function $- \log p(D|\theta)$ corresponds to the negative log likelihood of the model:

$$y = \hat{\theta}^\top x + \varepsilon, \; \varepsilon \sim \mathcal{N}(0, \sigma^2)?$$

  Why use the log?

Relationship to maximum likelihood estimation

- In Bayesian statistics, we define a **prior** distribution $p(\theta)$ over the parameter θ.
- By the Bayes rule, we can then define a **posterior** distribution $p(\theta|D)$ of θ :

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)} \propto p(D|\theta)p(\theta),$$

- We can then estimate θ by maximizing its **posterior** likelihood (MAP):

$$\hat{\theta}_{MAP} = \arg\max_{\theta} p(\theta|D)$$

Relationship to maximum likelihood estimation

- Maximizing the posterior likelihood yields

$$\begin{aligned}\hat{\theta}_{MAP} = \arg\max_{\theta} p(\theta|D) &= \arg\min_{\theta} -logp(\theta|D) \\ &= \arg\min_{\theta} -\log\left(p(D|\theta)p(\theta)\right) \\ &= \arg\min_{\theta} -\log p(D|\theta) - \log p(\theta).\end{aligned}$$

- We recover a penalized empirical risk minimization problem, where $\Omega(\theta) = -\log p(\theta)$.
- **Exercise :** what penalty do we get using the prior

$$\theta \sim \mathcal{N}(0, \sigma^2),$$

and which prior would lead to the ℓ_1 penalty?

Bayesian statistics

- In a purely Bayesian statistical framework, we would not look for the single value maximizing the posterior likelihood but rather consider distributions (over θ, over $\theta^\top x$...).
- This paradigm requires to know how to sample from the posterior distribution.

Relationship to maximum likelihood estimation

- Minimization of the penalized empirical risk can therefore be derived in the framework of likelihood maximization.
- Not necessary. Some loss functions (SVM) do not correspond to a negative log likelihood.
- Giving ourselves a model allows some type of theoretical analysis of our estimators: bias, consistency, admissibility...
- These analyses allow to understand the behavior of the estimators and to compare them, at the expense of some generality.
- This is useful, but it is important to keep in mind the sensitivity of the analysis to the assumptions made by the model, and the fact that in reality, the data was not generated by a model.

Penalized empirical risk minimization: summary

- Penalized empirical risk minimization allows us to **implement the idea of structural risk minimization**.
- **Lots** of penalties have been proposed, leading to various types of regularity for the estimators.
- Ideally, a good penalty corresponds to a prior for the estimator: we assume there exists a low risk function with this type of regularity.

Validation

Reminder: structural risk minimization

1. Define nested function sets of increasing complexity.
2. Minimize the empirical risk over each family.
3. **Choose the solution giving the best generalization performances.**

- Best generalization means lowest (population) risk

$$R(f) = \int_{\mathcal{X}\times\mathcal{Y}} L(y, f(x))d\mathbb{P} = \mathbf{E}[L(y, f(x))].$$

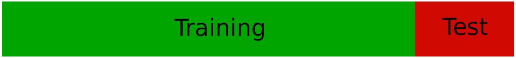
- But the very reason we need all this is that we don't have access to R!
- We need to estimate it as well.

Validation/Hold out procedure

- **[Exercise:]** why is empirical risk a poor estimator?

Validation/Hold out procedure

- **[Exercise:]** why is empirical risk a poor estimator?
- Alternative: split available data into training and test sets.

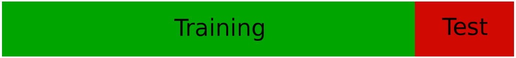


Training

Test

Validation/Hold out procedure

- **[Exercise:]** why is empirical risk a poor estimator?
- Alternative: split available data into training and test sets.



Training

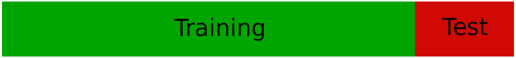
Test

- Formally:

$$\hat{R}^{\mathrm{HO}}\left(\hat{f}; D_n; I^{(t)}\right) = \frac{1}{n_v} \sum_{i \in D_n^{(v)}} L\left(y_i, \hat{f}_{D_n^{(t)}}(x_i)\right).$$

Validation/Hold out procedure

- **[Exercise:]** why is empirical risk a poor estimator?
- Alternative: split available data into training and test sets.



Training

Test

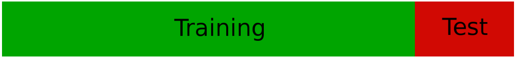
- Formally:

$$\hat{R}^{\mathrm{HO}}\left(\hat{f}; D_n; I^{(t)}\right) = \frac{1}{n_v} \sum_{i \in D_n^{(v)}} L\left(y_i, \hat{f}_{D_n^{(t)}}(x_i)\right).$$

- D_n: full set of n available data points. $I^{(t)}$: subset of indices used for training. $D_n^{(t)}$ (resp. $D_n^{(v)}$): set of data points restricted to training indices (resp. its complement).

Validation/Hold out procedure

- **[Exercise:]** why is empirical risk a poor estimator?
- Alternative: split available data into training and test sets.



Training Test

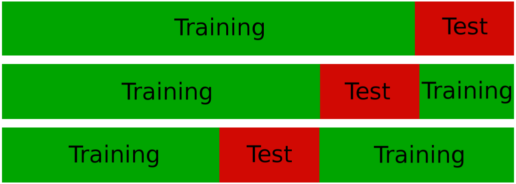
- Formally:

$$\hat{R}^{\mathrm{HO}}\left(\hat{f}; D_n; I^{(t)}\right) = \frac{1}{n_v} \sum_{i \in D_n^{(v)}} L\left(y_i, \hat{f}_{D_n^{(t)}}(x_i)\right).$$

- D_n: full set of n available data points. $I^{(t)}$: subset of indices used for training. $D_n^{(t)}$ (resp. $D_n^{(v)}$): set of data points restricted to training indices (resp. its complement).
- $\hat{f}$ denotes the learning algorithm whose risk we want to estimate. $\hat{f}_{D_n^{(t)}}$ is the function learnt by applying this algorithm to training data $D_n^{(t)}$.

Cross validation

- Idea: averaging several hold out estimators of the risk corresponding to different data splits:



Training Test

Training Test Training

Training Test Training

. . .

- Formally:

$$\hat{R}^{\mathrm{CV}}\left(\hat{f}; D_n; \left(I_j^{(t)}\right)_{1\leq j\leq B}\right) = \frac{1}{B}\sum_{j=1}^{B} \hat{R}^{\mathrm{HO}}\left(\hat{f}; D_n; I_j^{(t)}\right),$$

where $I_1^{(t)}, \ldots, I_B^{(t)}$ are non-empty proper subsets of $\{1, \ldots, n\}$.

Cross validation procedures

- CV estimators differ in how they define $I_1^{(t)}, \dots, I_B^{(t)}$.
- Most common: **V-fold CV**. Partition D_n into V sets of approximately equal cardinality $\frac{n}{V}$.
- **Leave-one-out** CV: V-fold with $V = n$.
- Monte-Carlo CV, leave-p-out CV...

Bias of the hold out estimator

- Hold out estimator: because training and validation samples are independent,

$$
\begin{aligned}
\mathbf{E}_{D_n\sim\mathcal{P}}\left[\hat{R}^{\mathrm{HO}}\left(\hat{f};D_n;I^{(t)}\right)\right] &= \frac{1}{n_v}\sum_{i\in D_n^{(v)}}\mathbf{E}_{(x_i,y_i)\cup D_n^{(t)}\sim\mathcal{P}}\left[L\left(y_i,\hat{f}_{D_n^{(t)}}(x_i)\right)\right]\\
&= \mathbf{E}_{(x,y)\cup D_n^{(t)}\sim\mathcal{P}}\left[L\left(y,\hat{f}_{D_n^{(t)}}(x)\right)\right]\\
&= \mathbf{E}_{D_n^{(t)}\sim\mathcal{P}}\left[\mathbf{E}_{(x,y)\sim\mathcal{P}}\left[L\left(y,\hat{f}_{D_n^{(t)}}(x)\right)\right]\right]\\
&= \mathbf{E}_{D_n^{(t)}\sim\mathcal{P}}\left[R\left(\hat{f}_{D_n^{(t)}}\right)\right].
\end{aligned}
$$

Bias of the hold out estimator

- Hold out estimator: because training and validation samples are independent,

$$\begin{aligned}\mathbf{E}_{D_n\sim\mathcal{P}}\left[\hat{R}^{\mathrm{HO}}\left(\hat{f};D_n;I^{(t)}\right)\right] &= \frac{1}{n_v}\sum_{i\in D_n^{(v)}}\mathbf{E}_{(x_i,y_i)\cup D_n^{(t)}\sim\mathcal{P}}\left[L\left(y_i,\hat{f}_{D_n^{(t)}}(x_i)\right)\right]\\ &= \mathbf{E}_{(x,y)\cup D_n^{(t)}\sim\mathcal{P}}\left[L\left(y,\hat{f}_{D_n^{(t)}}(x)\right)\right]\\ &= \mathbf{E}_{D_n^{(t)}\sim\mathcal{P}}\left[\mathbf{E}_{(x,y)\sim\mathcal{P}}\left[L\left(y,\hat{f}_{D_n^{(t)}}(x)\right)\right]\right]\\ &= \mathbf{E}_{D_n^{(t)}\sim\mathcal{P}}\left[R\left(\hat{f}_{D_n^{(t)}}\right)\right].\end{aligned}$$

- Only makes sense because $L\left(y_i,\hat{f}_{D_n^{(t)}}(x_i)\right)$ are i.i.d objects when (x_i,y_i) are independent of $\hat{f}_{D_n^{(t)}}$.

Bias of cross validation estimators

- For any cross validation estimator such that $\left|I_j^{(t)}\right| = n_t$,

$$\mathbf{E}_{D_n \sim \mathcal{P}} \left[\hat{R}^{\mathrm{CV}} \left(\hat{f}; D_n; I^{(t)} \right) \right] = \mathbf{E}_{D_n^{(t)} \sim \mathcal{P}} \left[R \left(\hat{f}_{D_n^{(t)}} \right) \right].$$

- The bias of such a CV estimator is therefore the difference between the risk expected using n and n_t training samples:

$$\mathrm{Bias} \left(\hat{R}^{\mathrm{CV}} \right) = \mathbf{E}_{D_n^{(t)} \sim \mathcal{P}} \left[R \left(\hat{f}_{D_n^{(t)}} \right) \right] - \mathbf{E}_{D_n \sim \mathcal{P}} \left[R \left(\hat{f}_{D_n} \right) \right].$$

- Usually non-negative (if $\hat{f}$ is a *smart rule*, *i.e.*, if its risk is a decreasing function of the size of the training set).
- More precise results for specific $\hat{f}$ and cross-validation estimators.

Bias of cross validation estimators

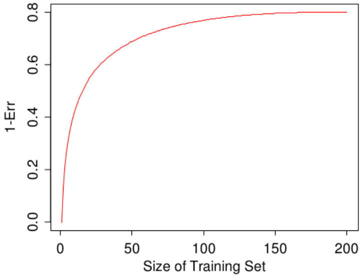


FIGURE 7.8. *Hypothetical learning curve for a classifier on a given task: a plot of* 1 − Err *versus the size of the training set* N*. With a dataset of* 200 *observations,* 5*-fold cross-validation would use training sets of size* 160*, which would behave much like the full set. However, with a dataset of* 50 *observations fivefold cross-validation would use training sets of size* 40*, and this would result in a considerable overestimate of prediction error.*

(from The Elements of Statistical Learning)

Variance of cross validation estimators

- All CV estimators with training sets of the same size n_t have the same bias. Difference of behavior explained by variances.
- For the hold out estimator,

$$\begin{aligned}\mathrm{Var}_{D_n}\left[\hat{R}^{\mathrm{HO}}\left(\hat{f};D_n;I^{(t)}\right)\right] =& \frac{1}{n_v}\mathbf{E}_{D_n^{(t)}}\left[\mathrm{Var}_{(x,y)}\left(L(y,\hat{f}_{D_n^{(t)}}(x))\right)\right] \\ &+ \mathrm{Var}_{D_n^{(t)}}\left[R\left(\hat{f}_{D_n^{(t)}}\right)\right].\end{aligned}$$

- First term: sensitivity of the error to a change of the validation sample. Also decreases in n_v (for fixed n_t).
- Second term: sensitivity of the risk to a change of the training set. Depends on stability of $\hat{f}$.

Variance of cross validation estimators

- No general result for CV.
- Less variable CV depends on framework (classification, regression, density estimation, model selection...).
- Factors of variability: n_v, n_t, B and stability of the algorithm.
- **What should I do**: no general answer, but it is standard to do 5 or 10 fold CV. Sometimes leave-one-out, but it is more expensive and known to often have large variance.
- More detailed answers in *A survey of cross-validation procedures for model selection* by S. Arlot and A. Celisse (from which this section is largely inspired).

Conditional vs expected prediction error

- There are actually two related quantities we could want to estimate:

$$R_{D_n} \triangleq \mathbf{E}_{(x,y)\sim\mathcal{P}}\left[L(y,\hat{f}_{D_n}(x))|D_n\right]$$
$$R \triangleq \mathbf{E}_{D_n,(x,y)\sim\mathcal{P}}\left[L(y,\hat{f}_{D_n}(x))\right] = \mathbf{E}_{D_n\sim\mathcal{P}}\left[R_{D_n}\right].$$

- Both can be useful depending on the context:
  1. are you always going to use this particular D_n
  2. or are you more interested in assessing the average performance of $\hat{f}$?
- CV empirically known to do a better job at estimating R than R_{D_n}. Actually hard to estimate R_{D_n} without using additional data. Keep that in mind if your objective is 1!

Model selection vs assessment

- Cross validation was historically first used for **model assessment**: estimate the generalization error of a given algorithm $\hat{f}$.

Model selection vs assessment

- Cross validation was historically first used for **model assessment**: estimate the generalization error of a given algorithm $\hat{f}$.
- As you will see in this class, there are lots of methods to solve the same inference problem. Most of them have options/hyperparameters (*e.g.*, penalized empirical risk minimization).

Model selection vs assessment

- Cross validation was historically first used for **model assessment**: estimate the generalization error of a given algorithm $\hat{f}$.
- As you will see in this class, there are lots of methods to solve the same inference problem. Most of them have options/hyperparameters (*e.g.*, penalized empirical risk minimization).
- We also need a tool for **model selection**: choose best method or best class of hypothesis for a particular problem.

Model selection vs assessment

- Cross validation was historically first used for **model assessment**: estimate the generalization error of a given algorithm $\hat{f}$.
- As you will see in this class, there are lots of methods to solve the same inference problem. Most of them have options/hyperparameters (*e.g.*, penalized empirical risk minimization).
- We also need a tool for **model selection**: choose best method or best class of hypothesis for a particular problem.
- Cross validation is commonly used for model selection as well. However, some **care** is necessary when doing both (which is often the case).

An example: selection bias in gene extraction

- 2002 paper by C. Ambroise and G. McLachlan: *Selection bias in gene extraction on the basis of microarray gene-expression data*.
- At the time, several paper using microarrays for cancer diagnosis claimed 0% generalization error estimated by cross validation:
  - Xiong et al.(2001), Mol Genet Metab, *Feature (Gene) Selection in Gene Expression-Based Tumor Classification*.
  - Zhang et al. (2001), Proc Natl Acad Sci USA, *Recursive partitioning for tumor classification with gene expression microarray data*.
  - Guyon et al. (2002), Mach Learn, *Gene Selection for Cancer Classification using Support Vector Machines*.

An example: selection bias in gene extraction

- General procedure was:
  1. Select a few genes which are good predictors over all samples.
  2. Perform cross validation to estimate the generalization error of a method using these genes.

  Exercise: What is wrong with this procedure? What should be done instead

An example: selection bias in gene extraction

- General procedure was:
  1. Select a few genes which are good predictors over all samples.
  2. Perform cross validation to estimate the generalization error of a method using these genes.

  Exercise: What is wrong with this procedure? What should be done instead
- The samples used to estimate the generalization error were used to select predictive genes.

An example: selection bias in gene extraction

- General procedure was:
  1. Select a few genes which are good predictors over all samples.
  2. Perform cross validation to estimate the generalization error of a method using these genes.

  Exercise: What is wrong with this procedure? What should be done instead
- The samples used to estimate the generalization error were used to select predictive genes.
- The predictive genes are optimal for the samples used to estimate the generalization error, which leads to an over-optimistic assessment regarding what will happen for actually new samples.

An example: selection bias in gene extraction

- General procedure was:
  1. Select a few genes which are good predictors over all samples.
  2. Perform cross validation to estimate the generalization error of a method using these genes.

  Exercise: What is wrong with this procedure? What should be done instead
- The samples used to estimate the generalization error were used to select predictive genes.
- The predictive genes are optimal for the samples used to estimate the generalization error, which leads to an over-optimistic assessment regarding what will happen for actually new samples.
- Picking a gene set is **model selection**, computing the generalization error of the estimator built over these genes is **model assessment**.

Model selection vs assessment

- Same thing goes for selecting a regularization parameter or a method: cross-validation error over D_n gives you an estimate of your best option (**model selection**), but it doesn't tell you how your best option will perform on new data (**model assessment**).

Model selection vs assessment

- Same thing goes for selecting a regularization parameter or a method: cross-validation error over D_n gives you an estimate of your best option (**model selection**), but it doesn't tell you how your best option will perform on new data (**model assessment**).
- **[Exercise:]** what would be an acceptable procedure to select a regularization parameter and estimate the resulting generalization error using a dataset D_n?

Model selection vs assessment

- Same thing goes for selecting a regularization parameter or a method: cross-validation error over D_n gives you an estimate of your best option (**model selection**), but it doesn't tell you how your best option will perform on new data (**model assessment**).
- **[Exercise:]** what would be an acceptable procedure to select a regularization parameter and estimate the resulting generalization error using a dataset D_n?
  - Train/Validation/Test split.
  - Double cross-validation.

Model selection vs assessment

- Principle: the data you use to estimate the generalization error of an algorithm cannot be used in any way to build the estimator. But it is easy to get confused, and difficult to strictly follow this principle when data is scarce.

Model selection vs assessment

- Principle: the data you use to estimate the generalization error of an algorithm cannot be used in any way to build the estimator. But it is easy to get confused, and difficult to strictly follow this principle when data is scarce.
- Maybe even more important than choosing best type of CV. Still results in many mistakes today.

Model selection vs assessment

- Principle: the data you use to estimate the generalization error of an algorithm cannot be used in any way to build the estimator. But it is easy to get confused, and difficult to strictly follow this principle when data is scarce.
- Maybe even more important than choosing best type of CV. Still results in many mistakes today.
- Other frequent source of mistakes in CV: duplicate/non i.i.d. samples.