Training with the full dataset after cross-validation?

Is it always a good idea to **train with the full dataset after cross-validation**? Put it another way, is it ok to train with **all** the samples in my dataset and **not** being able to check if this particular fitting **overfits**?

Some background on the problem:

Say I have a **family of models** parametrized by $\vec{\alpha}$. Say also that I have a set of N data points and that I do model selection with k-fold cross-validation to choose the model that best generalizes the data.

For model selection, I can do a search (e.g. a grid search) on $\vec{\alpha}$ by, for example, running k-fold cross-validation for each candidate. In each of the folds in cross-validation, I end up with **learned model** β_α .

The point of cross-validation is that for each of these folds I can check if the learned model had overfit, by testing it on "unseen data". Depending on the results, I could choose the model β_best learned for the parameters $\vec{\alpha}_\text{best}$ that generalized best during cross validation in the grid search.

Now, say that **after model selection**, I would like to use **all** the N points in my dataset and hopefully learn a better model. For this I could use the parameters $\vec{\alpha}_{best}$ corresponding to the model that I chose during model selection, and then after training on the full dataset, I would a get a **new** learned model β_{full} . The problem is that, if I use all points in my dataset for training, **I can't check if this new learned model β_{full} overfits** on any unseen data. What is the right way to think around this problem?

machine-learning cross-validation model-selection

edited Jan 30 '16 at 21:04
amoeba
35.6k 9 124 183

asked Jun 5 '11 at 16:50
Amelio Vazquez-Reina
4,310 6 39 72

Almost an exact duplicate: stats.stackexchange.com/questions/52274 with lots of worthy answers. Perhaps these threads should be merged but I am not sure in which direction. Both have accepted answers that are very good.
– amoeba Feb 11 '16 at 17:50

5 Answers

The way to think of cross-validation is as estimating the performance obtained using a method for building a model, rather than for estimating the performance of a model.

If you use cross-validation to estimate the hyperparameters of a model (the α s) and then use those hyper-parameters to fit a model to the whole dataset, then that is fine, provided that you recognise that the cross-validation estimate of performance is likely to be (possibly substantially) optimistically biased. This is because part of the model (the hyper-parameters) have been selected to minimise the cross-validation performance, so if the cross-validation statistic has a non-zero variance (and it will) there is the possibility of over-fitting the model selection criterion.

If you want to choose the hyper-parameters and estimate the performance of the resulting model then you need to perform a nested cross-validation, where the outer cross-validation is used to assess the performance of the model, and in each fold cross-validation is used to determine the hyper-parameters separately in each fold. You build the final model by using cross-validation on the whole set to choose the hyper-parameters and then build the classifier on the whole dataset using the optimized hyper-parameters.

This is of course computationally expensive, but worth it as the bias introduced by improper performance estimation can be large. See my paper

G. C. Cawley and N. L. C. Talbot, Over-fitting in model selection and subsequent selection bias in performance evaluation, Journal of Machine Learning Research, 2010. Research, vol. 11, pp. 2079-2107, July 2010. (www, pdf)

However, it is still possible to have over-fitting in model selection (nested cross-validation just allows you to test for it). A method I have found useful is to add a regularisation term to the cross-validation error that penalises hyper-parameter values likely to result in overly-complex models, see

G. C. Cawley and N. L. C. Talbot, Preventing over-fitting in model selection via Bayesian regularisation of the hyper-parameters, Journal of Machine Learning Research, volume 8, pages 841-861, April 2007. (www,pdf)

So the answers to your question are (i) yes, you should use the full dataset to produce your final model as the more data you use the more likely it is to generalise well but (ii) make sure you obtain an unbiased performance estimate via nested cross-validation and potentially consider penalising the cross-validation statistic to further avoid over-fitting in model selection.

edited Jan 28 '15 at 18:46
Scorchi ♦
19.8k 7 36 77

answered Apr 16 '12 at 13:56
Dikran Marsupial
29.6k 1 71 125

2 +1: Answers the question: "If you use cross-validation to estimate the hyperparameters of a model (the αs) and then use those hyper-parameters to fit a model to the whole dataset, then that is fine…" – Neil G Apr 19 '12 at 19:09

What could be a different approach from a nested cross-validation? Can we run some initial experiments and observe a proper setting for the hyper-parameter and then use cross-validation for the parameters? – soufanom Dec 15 '12 at 11:13

3 @soufanom, no, the use of "initial experiments" to make choices regarding the model is likely to result in over-fitting and almost certainly will introduce an optimistic bias into the performance analysis. The cross-validation used for performance analysis must repeat EVERY step used in fitting the model independently in each fold. The experiments in my paper show that kernel models can be very sensitive to this sort of bias, so it is vital to perform the model selection and performance evaluation with all possible rigour. – Dikran Marsupial Dec 15 '12 at 12:58

2 For kernel methods, such as the SVM, it is often possible to perform leave-one-out cross-validation at almost no computational cost (see the papers listed in my answer). I use this "virtual" leave-one-out cross-validation for tuning the hyper-parameters, nested in k-fold cross-validation for performance evaluation. The cost is then quite reasonable. In my opinion it is not acceptable to use any procedure where the performance evaluation is biased in any way by the tuning of the hyper-parameters. It is worth the computational expense to get a reliable estimate. – Dikran Marsupial Dec 15 '12 at 19:08

2 @DikranMarsupial. I don't quite get the third paragraph in your answer. If I do nested cross-validation, I will get a different set of hyperparameters for each fold of the **outer** CV (i.e. I get one set of hyperparameters from running the **inner** CV on a grid of parameters). How do I then choose the best set of hyperparameters? – Amelio Vazquez-Reina Jul 21 '13 at 1:31

|

Just to add to the answer by @mark999, Max Kuhn's `caret` package (Classification and Regression Training) is the most comprehensive source in R for model selection based on bootstrap cross validation or N-fold CV and some other schemes as well.

Not to disregard the greatness of the `rms` package, but `caret` lets you fit pretty much every learning method available in R, whereas `validate` only works with `rms` methods (I think).

The `caret` package is a single infrastructure to pre process data, fit and evaluate any popular model, hence it is simple to use for all methods and provides graphical assessment of many performance measures (something that next to the overfit problem might influence the model selection considerably as well) over your grid and variable importance.

See the package vignettes to get started (it is very simple to use)
Data Preprocessing
Variable Selection with caret
Model Building with caret
Variable Importance

You may also view the caret website for more information on the package, and specific implementation examples:
Offical caret website

edited Jan 28 '15 at 19:44
TLJ
563 3 11

answered Apr 14 '12 at 23:37
Momo
6,124 2 26 48

Thanks. Do you know if, after model selection (which is done by calling `train`), there is a way in caret to train with the full dataset? – Amelio Vazquez-Reina Feb 15 '13 at 20:59

Not sure if that is a good idea or why you would want to that, but you can just fit the final model returned by train to the full data set. – Momo Feb 17 '13 at 15:24

I believe that Frank Harrell would recommend bootstrap validation rather than cross validation. Bootstrap validation would allow you to validate the model fitted on the full data set, and is more stable than cross validation. You can do it in R using `validate` in Harrell's `rms` package.

See the book "Regression Modeling Strategies" by Harrell and/or "An Introduction to the Bootstrap" by Efron and Tibshirani for more information.

answered Jun 5 '11 at 19:13
mark999
2,240 2 15 26

8 To omit a next myth about "bad CV", this is a terminology problem -- Harrell's "cross validation" means N-fold CV, and "bootstrap validation" means resampling CV. Obviously I agree that this second flavor is more stable and overall nicer, but this is also a type of cross validation. – mbq Jun 5 '11 at 20:29

mark999 or @mbq, would you mind elaborating on how bootstrap would allow one to validate a model fitted on the full dataset? – Amelio Vazquez-Reina Feb 7 '13 at 19:19

1

@user27915816 Well, in principle nohow; the idea behind cross-validation is that you tests whether given *training method* is reliably making good models on a sets very similar to the final one, and, if so, generalise this observation to the full set with a silent assumptions that nothing strange will happen and that CV method you used is not somehow biased. This is of course almost always good enough, still you can never be *sure* that the model built on all the data you have is not overfitted. – mbq Feb 7 '13 at 23:10

I think you have a bunch of different questions here:

> The problem is that, if I use all points in my dataset for training, I can't check if this new learned model βfull overfits!

The thing is, you can use (one) validation step only for one thing: either for parameter optimization, (x)or for estimating generalization performance.

So, if you do parameter optimization by cross validation (or any other kind of data-driven parameter determination), you need test samples that are independent of those training and optimization samples. Dikran calls it nested cross validation, another name is double cross validation. Or, of course, an independent test set.

> So here's the question for this post : Is it a good idea to train with the full dataset after k-fold cross-validation? Or is it better instead to stick with one of the models learned in one of the cross-validation splits for αbest?

Using one of the cross validation models usually is *worse* than training on the full set (at least if your learning curve performance = f (nsamples) is still increasing. In practice, it is: if it wasn't, you would probably have set aside an independent test set.)

If you observe a large variation between the cross validation models (with the same parameters), then your models are unstable. In that case, aggregating the models can help and actually be better than using the *one* model trained on the whole data.

Update: This aggregation is the idea behind bagging applied to resampling without replacement (cross validation) instead of to resampling with replacement (bootstrap / out-of-bootstrap validation).

Here's a paper where we used this technique:
Beleites, C. & Salzer, R.: Assessing and improving the stability of chemometric models in small sample size situations, Anal Bioanal Chem, 390, 1261-1271 (2008).
DOI: 10.1007/s00216-007-1818-6

> Perhaps most importantly, how can I train with all points in my dataset and still fight overfitting?

By being very conservative with the degrees of freedom allowed for the "best" model, i.e. by taking into account the (random) uncertainty on the optimization cross validation results. If the d.f. are actually appropriate for the cross validation models, chances are good that they are not too many for the *larger* training set. The pitfall is that parameter optimization is actually multiple testing. You need to guard against accidentally good looking parameter sets.

edited Sep 14 '14 at 13:52 answered Apr 19 '12 at 19:30
cbeleites
16.8k 33 68

`...If you observe a large variation between the cross validation models (with the same parameters), then your models are unstable. In that case, aggregating the models can help...` Can you explain this a bit more? e.g. if I am running logistic regression in a 10-k cross validated setup and end up with 10 sets of coefficients, do you recommend aggregating the coeff estimates to form a final model? If so, how can this be done, just taking the means? – Berkmeister Aug 27 '14 at 10:12

@cbeleites can you elaborate on `If the d.f. are actually appropriate for the cross validation models`. In my understanding you are arguing that the train/validation sets are not very large when compared with the complete data set, am I right? – jpcgandre Sep 13 '14 at 7:09

1 @jpcgandre: Choosing one of the surrogate models for further use is in fact a data-driven model selection, which means that you need an outer independent level of validation. And in general, unless you have enough cases so you can actually do statistically meaningful model comparisons on the basis of testing $\frac{1}{k}$ of the total sample size, IMHO you should not select. – cbeleites Sep 14 '14 at 13:41

1 More importantly: the iterated cross validation surrogate models share the same set of hyperparameters. That is, they are equivalent in all you deem to be important but the arbitrary selection of training and test cases. Selecting a "good" model thus in fact should primarily select a good test/training set combination - which is fundamentally what we usually do *not* want: we want a choice that is generalizing well and thus not only working for favorable cases. From this point of view, selecting a surrogate model from a "normal" cross validation does not make any sense to me. – cbeleites Sep 14 '14 at 13:44

1 @jpcgandre: (d.f.) I argue that choosing a model complexity that is appropriate for training on $1 - \frac{1}{k}$ of the data set (which I argue is almost as large as the whole data set), you may arrive at a bias towards slightly too restrictive models for training on the whole data set. However, I don't think this should matter in practice, the more so as my impression in my field is that we rather tend to err towards too complex models. – cbeleites Sep 14 '14 at 13:55

|

What you do is not a cross validation, rather some kind of stochastic optimization.

The idea of CV is to simulate a performance on unseen data by performing several rounds of building the model on a subset of objects and testing on the remaining ones. The somewhat averaged results of all rounds are the **approximation of performance of a model trained on the whole set**.

In your case of model selection, you should perform a full CV for each parameter set, and thus get a on-full-set performance approximation for each setup, so seemingly the thing you wanted to have.

However, note that it is not at all guaranteed that the model with best approximated accuracy will be the best in fact -- you may cross-validate the whole model selection procedure to see that there exist some range in parameter space for which the differences in model accuracies are not significant.

answered Jun 5 '11 at 20:49

mbq
18.5k 8 53 108

2 Thanks @mbq, but I'm not sure I follow. I **do** N-fold cross-validation for each point value of my grid search in the hyperparameter space. The average result of the N-folds gives me the approximation that you mention, which I use to compare models and do model selection by selecting the model the best fits the validation set. My question is about what happens when I train with the full dataset. I think the learned model changes (the $\vec{\beta}$ parameters of the learned model change), and in principle I have no way to know if I suffer from overfitting. – Amelio Vazquez-Reina Jun 5 '11 at 22:05

@AmV If so, ok -- as I wrote, CV already tests full set scenario, you cannot say more without new data. Again, you can at most do a nested CV to see if there is no overfitting imposed by the model selection itself (if the selection gives very good improvement or the data is noisy the risk of this is quite big). – mbq Jun 5 '11 at 22:15