Quora Search Home Dee Add Question

Machine Learning: Logistic Regression Regression (statistics) Statistics (academic discipline) scikit-learn Edit

What is the difference between L1 and L2 regularization?

Edit

What is the most simplest explanation of L1 and L2 regularization.
How does it solves the problem of overfitting?
Which regularizer to use when?

Edit

Comment · Share (3) · Report · Options

10 Answers Ask to Answer

Dee Pypunk Add Bio · Make Anonymous

Add your answer, or answer later.

55

Aleks Jakulin, 10 years in machine learning

Votes by **Alexander Blocker**, Statistician at Google, PhD in statistics from ..., **Don van der Drift**, In PhD Physics program for 2.5 years at Technis..., Zack Raymond Black, Joseph Quattrocchi, and 50 more.

When you have lots of parameters but not enough data points, regression can overfit. For example, you might find that logistic regression proposing a model fully confident that all patients on one side of the hyperplane will die with 100% probability and the ones on the other side will live with 100% probability.

Now, we all know that this is unlikely. In fact, it's pretty rare that you'd ever have an effect even as strong as smoking. Such egregiously confident predictions are associated with high values of regression coefficients. Thus, regularization is about incorporating what we know about regression and data on top of what's actually in the available data: often as simple as indicating that high coefficient values need a lot of data to be acceptable.

The Bayesian regularization paradigm assumes what a typical regression problem should be like - and then mathematically fuses the prior expectations with what's fit from the data: understanding that there are a number of models that could all explain the observed data. Other paradigms involve ad-hoc algorithms or estimators that are computationally efficient, sometimes have bounds on their performance, but it's less of a priority to seek a simple unifying theory of what's actually going on. Bayesians are happy to employ efficient ad-hoc algorithms understanding that they are approximations to the general theory.

Two popular regularization methods are L1 and L2. If you're familiar with Bayesian statistics: L1 usually corresponds to setting a Laplacean prior on the regression coefficients - and picking a maximum a posteriori hypothesis. L2 similarly corresponds to Gaussian prior. As one moves away from zero, the probability for such a coefficient grows progressively smaller.

Follow Question Promote Question

Related Questions

- Machine Learning: How do you decide to regularize between L1/L2 or best/greedy subset selection?
- Machine Learning: Are there any non-R statistical packages that perform L1/L2 regularization for regression modeling?
- Is that OK to use Newton's method on some variables(L2 regularization) meanwhile use iterative thresholding method on othe... (continue)
- Mathematics: What is a good way to find convex conjugate of l1 and l2 norm?
- In scikit-learn logistic regression, what are l1 and l2 values?

More Related Questions

Share Question

Twitter Facebook

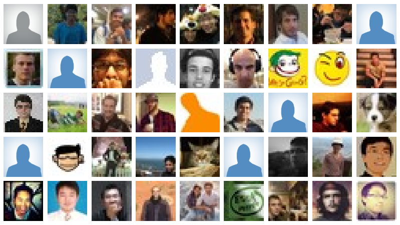
Question Stats

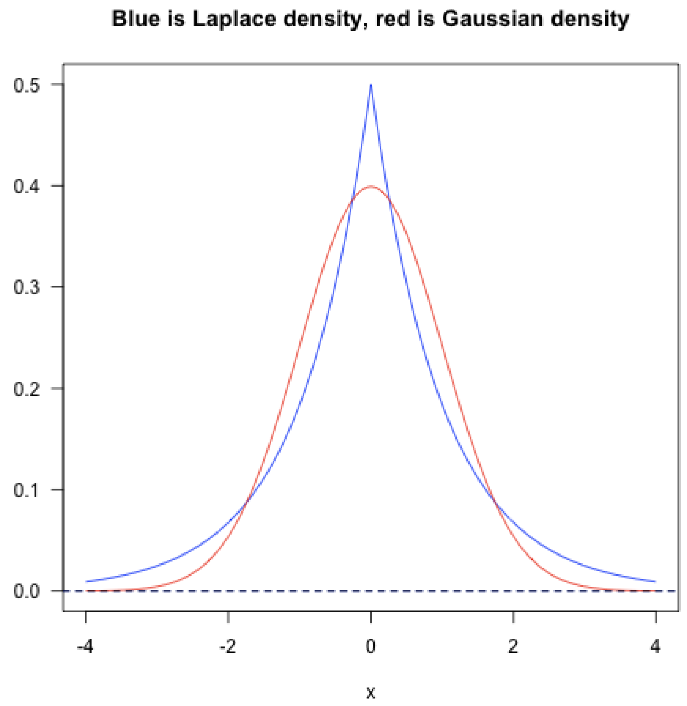
Latest activity 11 Oct

This question has **2** monitors with **242862** topic followers.

10,211 views on this question.

104 people are following this question.





As you can see, L1/Laplace tends to tolerate both large values as well as very small values of coefficients more than L2/Gaussian (tails).

Regularization works by adding the penalty associated with the coefficient values to the error of the hypothesis. This way, an accurate hypothesis with unlikely coefficients would be penalized whila a somewhat less accurate but more conservative hypothesis with low coefficients would not be penalized as much.

For more information and evaluations, see http://www.stat.columbia.edu/~ge... - I personally prefer Cauchy priors, which correspond to log(1+L2) penalty/regularization terms.

1+ Comments · Share (3) · Thank · Report · 19 Sep, 2012

26

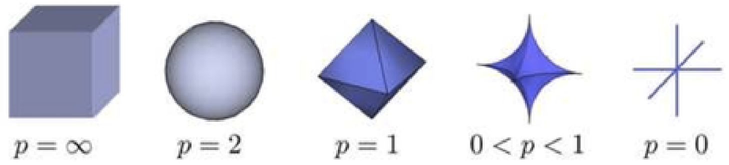
Manjari Narayan, Rice Ph.D. candidate in Signal Processing, but specializing in Statistics

Votes by **Justin Rising**, PhD in statistics with a dissertation in probab..., **Giri Gopalan**, Ph.D. student in Statistics at Harvard University, **Jack Rae**, Quora Data Scientist, Charles H Martin, and 21 more.

Justin Solomon has a great answer on the difference between L1 and L2 norms and the implications for regularization.

***ℓ1* vs *ℓ2* for signal estimation:**
Here is what a signal that is sparse or approximately sparse i.e. that belongs to the ell-1 ball looks like. It becomes extremely unlikely that an ***ℓ2*** penalty can recover a sparse signal since very few solutions of such a cost function are truly sparse. ***ℓ1*** penalties on the other hand are great for recovering truly sparse signals, as they are computationally tractable but still capable of recovering the exact sparse solution. ***ℓ2*** penalization is preferable for data that is not at all sparse, i.e. where you do not expect regression coefficients to show a decaying property. In such cases, incorrectly using an ***ℓ1*** penalty for non-sparse data will give you give you a large estimation error.

Figure: ℓp ball. As the value of p decreases, the size of the corresponding ℓp space also decreases. This can be seen visually when comparing the the size of the spaces of signals, in three dimensions, for which the ℓp norm is less than or equal to one. The volume of these ℓp "balls" decreases with p. [2]



***ℓ1* vs *ℓ2* for prediction:**
Typically ridge or ***ℓ2*** penalties are much better for minimizing prediction error rather than ***ℓ1*** penalties. The reason for this is that when two predictors are highly correlated, ***ℓ1*** regularizer will simply pick one of the two predictors. In contrast, the ***ℓ2*** regularizer will keep both of them and jointly shrink the corresponding coefficients a little bit. Thus, while the ***ℓ1*** penalty can certainly reduce overfitting, you may also experience a loss in predictive power.

A Clarification on *ℓ1*-regularization for Exact Sparse Signal Recovery:
However I want to comment on a frequently used analogy that ***ℓ1***-regularization is *equivalent* to MAP estimation using Laplacian priors. The notion of equivalence here is very subtle.

Remember if the true signal is sparse its coefficients have exactly k non-zeros or and approximately sparse if k really large coefficients and with the rest of the coefficients decaying to zero quickly. ***ℓ1*** regularization doesn't merely encourage sparse solutions, but is capable of exactly recovering a signal that is sparse.

Between 1999-2005, many exciting results in statistics and signal processing [3-6] demonstrated that if the underlying signal was extremely sparse and the design matrix satisfied certain conditions the solution to ***ℓ1***-regularized objective would coincide with the ***ℓ0***-regularized (best subset selection) objective, despite having an overall under-determined and high dimensional problem. This would not be possible with ***ℓ2*** regularization.

An analogous question when performing MAP estimation using laplacian priors would be,

"What class of signals does such a cost function recover accurately ?"

The bottom line here is that geometric intuition that ***ℓ1***-regularization is *like* laplacian regularized MAP does not mean that laplacian distributions can be used to describe sparse or compressible signals.

A recent paper by Gribonval, et al. [1] demonstrated the following

> many distributions revolving around maximum a posteriori (MAP) interpretation of sparse regularized estimators are in fact incompressible, in the limit of large problem sizes. We especially highlight the Laplace distribution and ℓ1 regularized estimators such as the Lasso and Basis Pursuit denoising. We rigorously disprove the myth that the success of ℓ1 minimization for compressed sensing image reconstruction is a simple corollary of a Laplace model of images combined with Bayesian MAP estimation, and show that in fact quite the reverse is true.

This paper [1] proves that many instances of signals drawn from a laplacian distribution are simply not sparse enough to be good candidates for l1 like recovery. In fact such signals are better estimated using ordinary least squares! An illustration of Fig. 4 from the paper is provided below.

Update: All the theoretical results show that sparse or approximately sparse signals can be recovered effectively by minimizing an ***ℓ1***-regularized cost function. But you cannot assume that just because laplacian priors have a "sparsifying" property when used in a cost function that one can use the same distribution as a generative model for the signal.

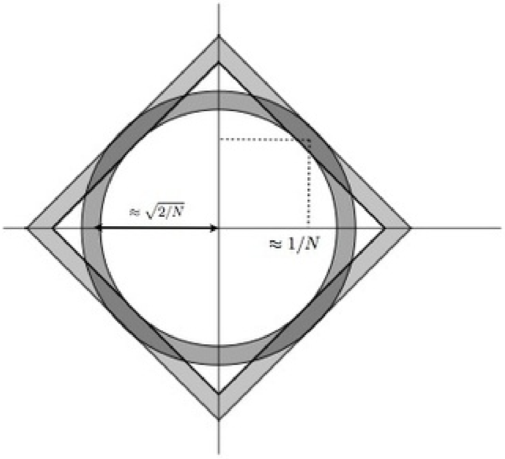


Fig. 4. A cartoon view of the ℓ^1 and ℓ^2 "rings" where vectors with iid Laplace-distributed entries concentrate. The radius of the ℓ^2 ring is of the order of $\sqrt{2/N}$ while that of the ℓ^1 ring is one, corresponding to vectors with flat entries $|\mathbf{x}|_n \approx 1/N$.

Aleks Jakulin pointed in the comments, that it is not a standard assumption in Bayesian statistics to assume that the data is drawn from the prior. While this maybe true, this result was an important clarification for quasi-bayesians who strongly care about the equivalence of ***ℓ0-ℓ1*** solutions in signal processing and communication theory—That the **theoretical results for exact recovery of sparse signals do not apply if you assume that the geometric intuition of the compressible signal belonging to the l1-ball (see below) is equivalent to probabilistic or generative model interpretation that the signal as iid laplacian.**

[1] http://arxiv.org/pdf/1102.1249v3...
[2] Compressible signals
[3] Compressed sensing
[4]Uncertainty principles and ideal atomic decomposition
[5]Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information

2+ Comments • Share (2) • Thank • Report • 14 Sep

49

YunFang Juan, Applied Machine Learning to Yahoo! Search and Facebook Ads.

Votes by **Jack Rae**, Quora Data Scientist, **Yair Livne**, Econ PhD from Stanford, took 2 years of stats P..., Tao Xu, Andrew Tulloch, and 44 more.

Practically, I think the biggest reasons for regularization are 1) to avoid overfitting by not generating high coefficients for predictors that are sparse. 2) to stabilize the estimates especially when there's collinearity in the data.

1) is inherent in the regularization framework. Since there are two forces pulling each other in the objective function, if there's no meaningful loss reduction, the increased penalty from the regularization term wouldn't improve the overall objective function. This is a great property since a lot of noise would be automatically filtered out from the model.

To give you an example for 2), if you have two predictors that have same values, if you just run a regression algorithm on it since the data matrix is singular, your beta coefficients will be Inf if you try to do a straight matrix inversion. But if you add a very small regularization lambda to it, you will get stable beta coefficients with the coefficient values evenly divided between the equivalent two variables.

For the difference between L1 and L2, the following graph demonstrates why people bother to have L1 since L2 has such an elegant analytical solution and is so computationally straightforward. Regularized regression can also be represented as a constrained regression problem (since they are Lagrangian equivalent). In Graph (a), the black square represents the feasible region of of the L1 regularization while graph (b) represents the feasible region for L2 regularization. The contours in the plots represent different loss values (for the unconstrained regression model). The feasible point that minimizes the loss is more likely to happen on the coordinates on graph (a) than on graph (b) since graph (a) is more **angular.** This effect amplifies when your number of coefficients increases, i.e. from 2 to 200.

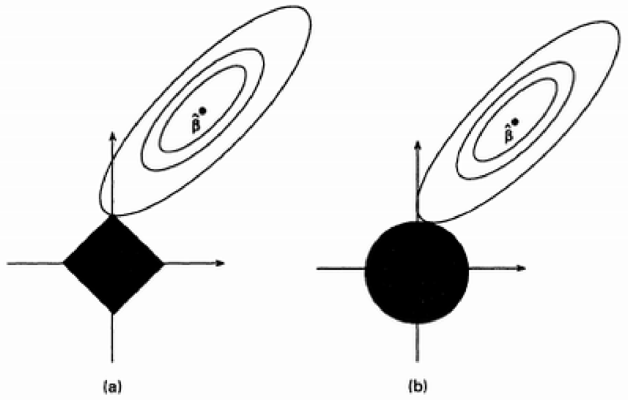


Fig. 2. Estimation picture for (a) the lasso and (b) ridge regression

The implication of this is that the L1 regularization gives you **sparse** estimates. Namely, in a high dimensional space, you got mostly zeros and a small number of non-zero coefficients. This is huge since it incorporates variable selection to the modeling problem. In addition, if you have to score a large sample with your model, you can have a lot of computational savings since you don't have to compute features(predictors) whose coefficient is 0. I personally think L1 regularization is one of the most beautiful things in machine learning and convex optimization. It is indeed widely used in bioinformatics and large scale machine learning for companies like Facebook, Yahoo, Google and Microsoft.

Comment · Share (2) · Thank · Report · 26 Sep

5 **Rick Barber, Grad student doing data things**

Votes by Jofin Joseph, Rishabh Sharma, Iain Marshall, and Srikanth Iyer.

Let me also suggest reading this paper by Andrew Ng. It basically says that for a number of methods, if you expect to have a high number of irrelevant features, you will need fewer training examples to generalize well with L1 compared to L2

http://www-cs.stanford.edu/peopl...

Abstract:

We consider supervised learning in the presence of very many irrelevant features, and study two different regularization methods for preventing overfitting. Focusing on logistic regression, we show that using L1 regularization of the parameters, the sample complexity (i.e., the number of training examples required to learn "well,") grows only logarithmically in the number of irrelevant features. This logarithmic rate matches the best known bounds for feature selection, and indicates that L1 regularized logistic regression can be effective even if there are exponentially many irrelevant features as there are training examples. We also give a lower-bound showing that any rotationally invariant algorithm—including logistic regression with L2 regularization, SVMs, and neural networks trained by backpropagation—has a worst case sample complexity that grows at least linearly in the number of irrelevant features.

1 Comment · Share · Thank · Report · 2 Jan

2 **C.V. Krishnakumar, तत् त्वम् असि**

Vote by Srikanth Iyer.

These slides might help : http://cseweb.ucsd.edu/~elkan/25...

Comment · Share · Thank · Report · 19 Sep, 2012

22

Justin Solomon, PhD student, Computer Science

Votes by Abhishek Shivkumar, Sumit Bam Shrestha, Salil P Navgire, Luke Chen, and 17 more.

There are many ways to understand the need for and approaches to regularization. I won't attempt to summarize the ideas here, but you should explore statistics or machine learning literature to get a high-level view. In particular, you can view regularization as a prior on the distribution from which your data is drawn (most famously Gaussian for least-squares), as a way to punish high values in regression coefficients, and so on. I prefer a more naive but somewhat more understandable (for me!) viewpoint.

Let's say you wish to solve the linear problem $Ax = b$. Here, A is a matrix and b is a vector. We spend lots of time in linear algebra worrying about the *exactly-* and *over-determined* cases, in which A is at least as tall as it is wide, but instead let's assume the system is *under-determined*, e.g. A is wider than it is tall, in which case there generally exist infinitely many solutions to the problem at hand.

This case is troublesome, because there are **multiple** possible x's you might want to recover. To choose one, we can solve the following optimization problem:

> MINIMIZE $||x||$ WITH RESPECT TO $Ax = b$

This is called the **least-norm solution**. In many ways, it says "In the absence of any other information, I might as well make x small."

But there's one thing I've neglected in the notation above: The norm $||x||$. It turns out, this makes all the difference!

In particular, consider the vectors $a = (0.5, 0.5)$ and $b = (-1, 0)$. We can compute two possible norms:

- $||a||_1 = |0.5| + |0.5| = 1$ and $||b||_1 = |-1| + |0| = 1$
- $||a||_2 = \sqrt{0.5^2 + 0.5^2} = 1/\sqrt{2} < 1$ and $||b||_2 = \sqrt{(-1)^2 + (0)^2} = 1$

So, the two vectors are equivalent with respect to the L1 norm but different with respect to the L2 norm. This is because **squaring a number punishes large values more than it punishes small values**.

Thus, solving the minimization problem above with $||x||_2$ (so-called "Tikhonov regularization") *really* wants small values in all slots of x, whereas solving the L1 version doesn't care if it puts all the large values into a single slot of x.

Practically speaking, we can see L2 regularization spreads error throughout the vector x, whereas L1 is happy (in many cases) with a *sparse* x, meaning that some values in x are **exactly** zero while others may be relatively large. The former case is sufficient and indeed suitable for a variety of statistical problems, but the latter is gaining traction through the field of compressive sensing. From a non-rigorous standpoint, compressive sensing assumes not that observations come from Gaussian-distributed sources about ground truth but rather that sparse or simple solutions to equations are preferable or more likely (the "Occam's Razor" approach).

2 Comments · Share · Thank · Report · 24 Nov, 2012

5

Eren Golge, CS enthusiast,MS Student in Bilkent University.

Votes by Mayank Gupta, John McGonagle, Arun Venkatraman, and Kevin Wang.

L1 is the first moment norm |x1-x2| (|w| for regularization case) that is simply the absolute dıstance between two points where L2 is second moment norm corresponding to Eucledian Distance that is |x1-x2|^2 (|w|^2 for regularization case) .

Another big difference is L1 is not differentiable thus for finding weigths that minimize your error function, you cannot use gradient based approaches. L2 give you a regularization term that is differentiable so it is suitable to be used with gradient descent.

you may find some more from this slide :http://cs.nyu.edu/~rostami/prese...

Comment · Share · Thank · Report · 21 Nov, 2012

6

Ethan Richman Suggest Bio

Votes by James Pirruccello, Arun Venkatraman, Rishabh Sharma, Srikanth Iyer, and 1 more.

L1 and L2 regularization add a cost to high valued weights to prevent overfitting. L1 regularization is an absolute value cost function and tends to set more weights to 0 (places more mass on zero weights) compared to L2 regularization.

Comment · Share · Thank · Report · 19 Sep, 2012

2

Charles H Martin, Consultant, I predict things

Vote by Arun Nedunchezhian.

The L1 Norm provides a near optimal sparse solution when the underlying signal /data is sparse in some (say overcomplete) basis and the signal to noise ratio (SNR) is high

The L2 Norm is suitable for non-sparse solutions and/or bandwidth limited signals

For example, a face is sparse in a Wavelet / Harr basis; this is the basis of the JPEG 2000 algorithm

A document, however, is generally not that sparse in the bag of words representation, and L2 methods can work very well here

Notice I did not say anything about the noise distribution. Also, the optimal sparse solution is the L0 norm, not the L1 norm.

For more details, see my blog
https://charlesmartin14.wordpres...

(note: the bounding theorems by Tao et. al. essentially say the same thing as Ng's paper--you need fewer examples when the system is sparse)

Comment · Share · Thank · Report · 2 Jan

3

Felix Heide, PhD candidate

Votes by Charles H Martin and Michael Bailey.

Here is an interesting paper on L_2 vs. L_1 minimization, I just read:
http://www.cs.ubc.ca/~ascher/pap...

Comment · Share · Thank · Report · 4 Nov, 2012

Dee Pypunk Add Bio · Make Anonymous

Add your answer, or answer later.