Lecture 3: feature selection and regularization

wbg231

December 2022

1 complexity of the hypothesis space

- there is a trade off between the size of the hypothesis space and over fitting (bias variance trade off)
- general approach to control complexity
  1. learn a sequence of models varying in complexity from the training data complexity
     $$F_1 \subseteq F_2 \cdots \subseteq F_n$$
     that is each model will expand the hypothesis space slightly we could do this for instance with $\mathcal{F} = \{$all polynomial functions $\}$ and then $\mathcal{F}_d = \{$all polynomials of degree $\leq d\}$
  2. then we select one model based on some score

linear regression example

- suppose the problem of picking the optimal number of linear features $\mathcal{F}_d = \{$a linear function using less than d features$\}$
- in this case we have $2^|d|$ possible combinations if d is the total number of features
- so this becomes a subset selection problem
- we could do this in a greedy forward method that is start with an empty model with no features, then check every feature not yet in our model learn a with all features already in the model plus the new feature and check its score then. find the best scoring model and if it improves the best score of our model at last step add that parameter to the model and start over, otherwise end
- that was a lot longer to write out in words than math

l1 and l2 regularization

complexity penalty

- an objective with a complexity penalty can be written as

$$j(\theta) = \ell(\theta) + \lambda R(|\theta|)$$

where R is some regularization function that penalizes the model for the magnitude of the features and λ is a regularization coefficient

- the constrained ERN problem is thus given by

$$min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \ell(f_\theta(x_i), y_i) + \lambda R(\theta)$$

- we prefer a smaller parameter because if we push the estimated weights to be small re-estimating them on a new dataset would likely not cause a dramatic change (and thus the model is resistant to overfitting)

l2 penalty

- ridge regression objective is given by

$$j(\theta) = ||Xw||_2^2 + \lambda ||w||_2^2$$

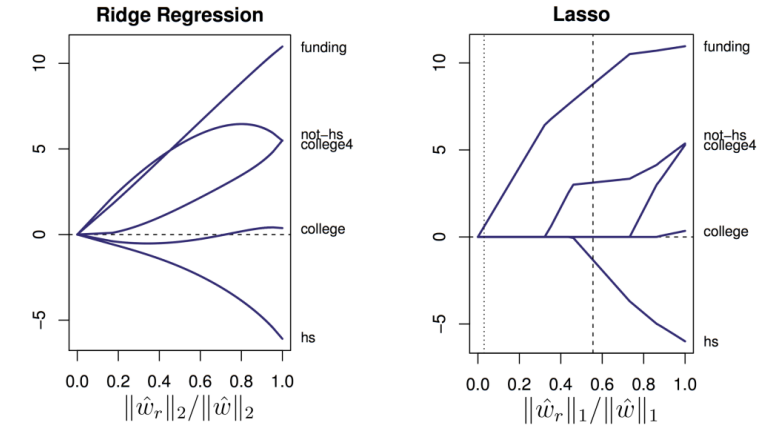
where $||w||_2^2 = w_1^2 + ...w_d^2$ is the square of the ℓ_2 norm

- can add an ℓ_2 normalization penalty to other models as well
- if f(x)=$w^t x$ is Lipschitz continuous with Lipschitz constant $L = ||w||_2^2$ then when moving from x to x+h is bounded by $L||h|| = ||w||_2||h||$
- $|f(x+h) - f(x)| = |w^t(x+h) - w^T x| = |w^t h| \leq ||w||_2||h||_2$ (this holds in general for any c norm)
- the point is lowering the norm of weights reduces the max rate our optimal function f can change at
- ridgre regression closed form is always well defined because $(X^T X + \lambda I)$ is always invertible
- the ℓ_1 or lasso objective can be defined analogously as

$$j(\theta) = \theta^t w + \lambda |\theta|_1$$

where $||w||_1 = \sum_i |w_i|$

- bellow is a graph of the regularization charts of ridge and lasso regression



- so first of all keep in mind that a large value of $||w_r||_k$ implies a low λ so in ridge regression we reduce parameters based on how much changing them would effect square loss so will only lower a parameter until lowering another parameter reduces loss more
- lasso on the other hand has constant reduction in loss from lowering a parameter regardless of it's value, so it will reduce one parameter at a time to zero
- so lasso likes a spare solution
- sparse solutions can be good for the following reasons
  1. we do not that parameter in our prediction so computation becomes cheaper
  2. it takes less memory to store our features
  3. it is easier to identify whcih features are really importnat when there are only a few
  4. and hte prediction function may generalize better

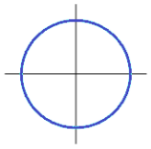
why does l1 regularization lead to sparse solutions

- constrained erm

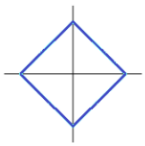
$$\hat{w} = argmin_{||W||_1 \leq r} ||Xw - y||_2^2$$

- can also be written in terms of penalized erm in practice both are usually effective
- so if we look at the constrained set for $\lambda = r$ constant in constrained erm we have

- ℓ_2 contour: $w_1^2 + w_2^2 = r$

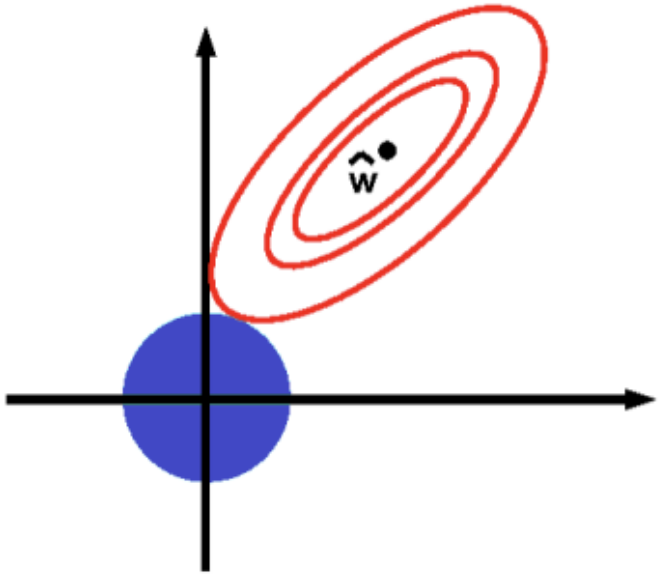


- ℓ_1 contour: $|w_1| + |w_2| = r$

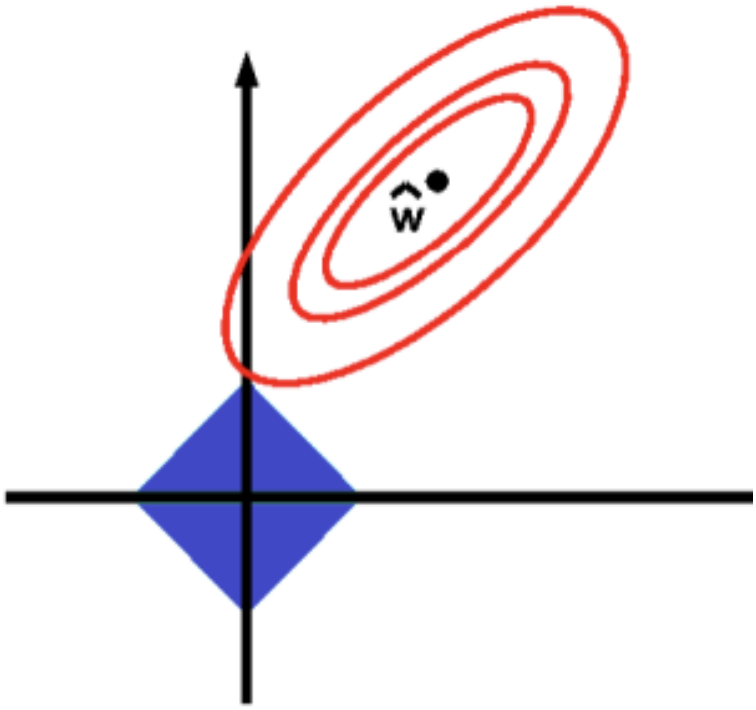


-

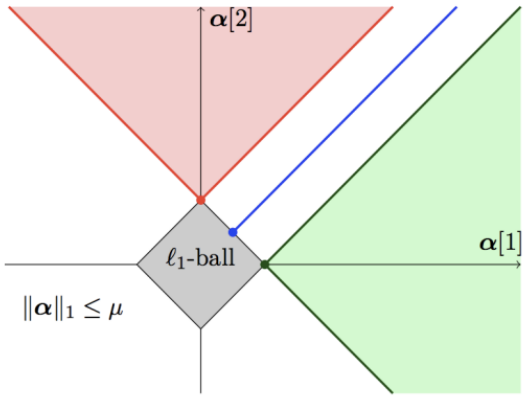
- then we can compare this to the continuous of our objective



- we can compare this with the lasso constrained erm

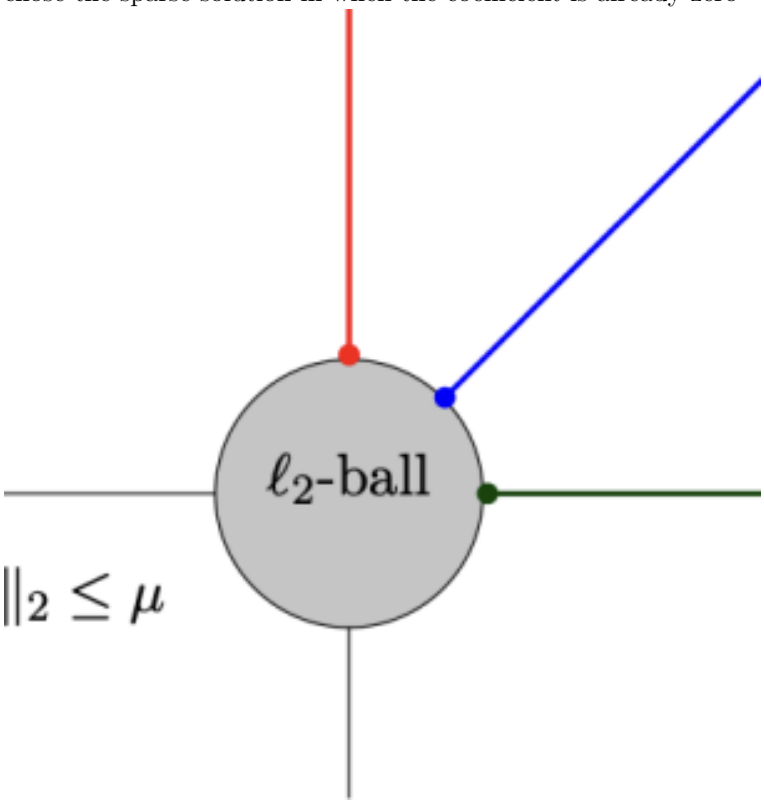


- notice that the set of points that are more likely to be close to the corner of the square (ie spare lasso solution is quite large) as can be seen from this diagram



ling for Image and Vision Processing **Fig 1.6**

- we can compare this to constrained ℓ_2 regularization and which will only chose the sparse solution in when the coefficient is already zero



minimizing the lasso objective

- recall that we can write that our lasso objective $j(w) = ||Xw-y||_2^2+\lambda||w||_1$ is not differentiable
- so we can deal with this in a number of ways
- we can either use coordinate descent, linear programmings or projected stochastic gradient descent