

CSE 6740 Lecture 6

How Do I Predict a Continuous Variable? (Regression)

Alexander Gray

agray@cc.gatech.edu

Georgia Institute of Technology

Today

1. A bit more about cross-validation and KDE
2. Regression (*How can I predict a continuous variable?*)

A bit more about cross-validation

A few more things to say...

Using Cross-Validation

Cross-validation is used to estimate the generalization error for different model classes – but we had to hold out some of the data from training to make the estimate.

After we are done selecting a model class, we can go back and use *all* the training data to train a final model. So cross-validation is ultimately still just a way of obtaining a relative *score* for different model classes.

Using Cross-Validation

Cross-validation can be used for any loss function; we just have to be able to evaluate the loss on *out-of-sample* data (points not in the training set).

Since this is true of loss functions typical of unsupervised learning (such as the likelihood) as well as of supervised learning, we can use cross-validation for unsupervised learning, as long as there is a loss function.

KDE: Likelihood CV

Consider using likelihood loss for KDE. Leave-one-out likelihood cross-validation maximizes

$$\text{CV}_l(h) = \frac{1}{N} \sum_{i=1}^N \log \hat{f}_h^{-i}(x_i). \quad (1)$$

Under what turn out to be strong assumptions on the density and kernel, $-\text{CV}_l(h)$ is, up to a constant, an unbiased estimator of the expected KL error.

However, it can be shown that for “typical” pdf’s, the non-robustness of the likelihood makes it inconsistent.

KDE: Least-Squares CV

Now we'll consider the *mean integrated squared error* (MISE) as the loss. We want the value of h that minimizes it. It can be written

$$\int \left(\hat{f}_h - f \right)^2 = \int \hat{f}_h^2 - 2 \int \hat{f}_h f + \int f^2. \quad (2)$$

The first term can be obtained analytically. The last term doesn't depend on h . For the second term we have

$$\mathbb{E} \int \hat{f}_h(x) f(x) dx = \mathbb{E} \int \hat{f}_h^{-N}(x) f(x) dx \quad (3)$$

$$= \mathbb{E} \hat{f}_h^{-N}(x) \quad (4)$$

$$= \mathbb{E} \frac{1}{N} \sum_{i=1}^N \hat{f}_h^{-i}(x). \quad (5)$$

KDE: Least-Squares CV

This motivates

$$\mathbf{CV}_{L_2}(h) = \int \widehat{f}_h^2 - 2 \frac{1}{N} \sum_{i=1}^N \widehat{f}_h^{-i}(x_i) \quad (6)$$

since $\mathbb{E}\mathbf{CV}_{L_2}(h) + \int f^2 = \mathbb{E}\mathbf{MISE}(\widehat{f}_h)$, an unbiased estimator up to a constant.

It can be shown under mild assumptions that

$$\frac{\mathbf{MISE}_{\mathbf{CV}_{L_2}}(\{x\}_N)}{\mathbf{MISE}^*(\{x\}_N)} \rightarrow 1 \quad \text{as } N \rightarrow \infty. \quad (7)$$

Convergence Rate Lower Bound

Let \mathcal{F} be the set of all PDF's and let $f^{(m)}$ denote the m^{th} derivative of f . Define

$$\mathcal{F}_m(c) = \left\{ f \in \mathcal{F} : \int |f^{(m)}(x)|^2 dx \leq c^2 \right\}. \quad (8)$$

For *any* estimator \hat{f}_N ,

$$\sup_{f \in \mathcal{F}_m(c)} \mathbb{E}_f \int (\hat{f}_N(x) - f(x))^2 dx \geq b \left(\frac{1}{N} \right)^{2m/(2m+1)} \quad (9)$$

where $b > 0$ is a universal constant that depends only on m and c . Plugging in $m = 2$ yields $O(N^{-4/5})$.

KDE Optimal Kernel Function

We can also see what kernel function K minimizes the risk. If we do this we obtain this kernel:

$$K(x) = \frac{3}{4} (1 - x^2) I(|x| \leq 1). \quad (10)$$

This is called the *Epanechnikov kernel*.

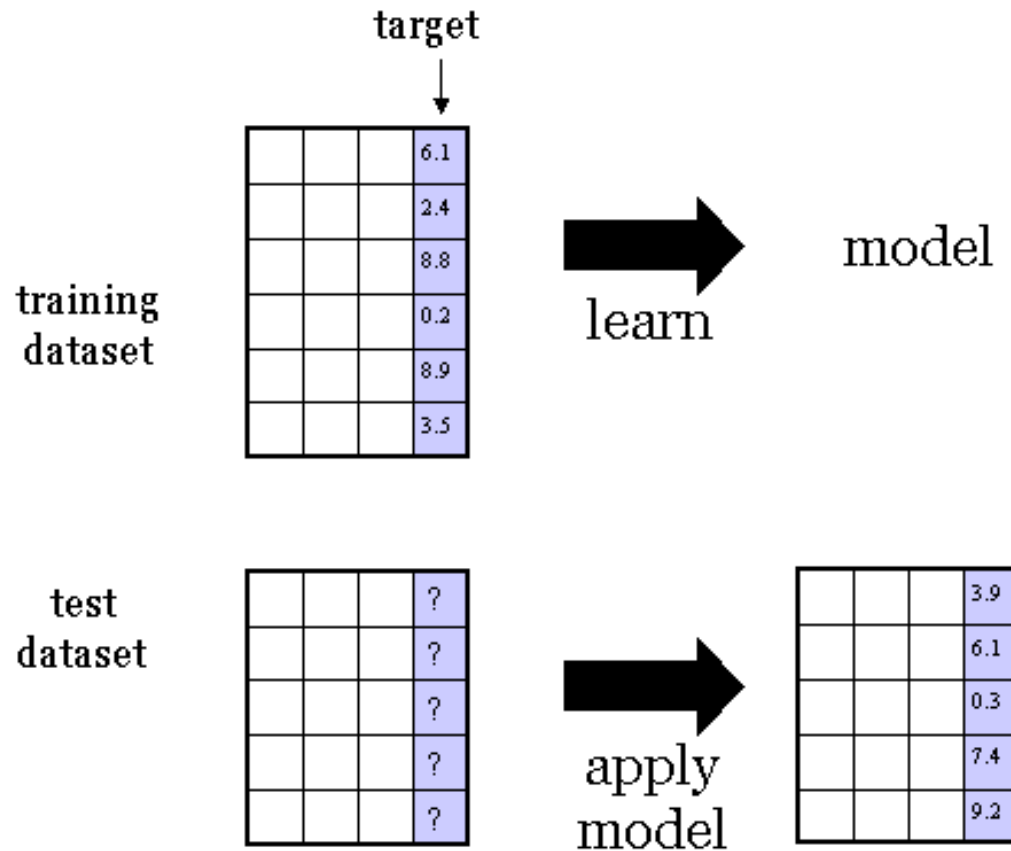
Thus the Gaussian kernel is not the optimal kernel.

In practice the choice of bandwidth is much more important than the choice of kernel function.

Regression

How can we predict a continuous variable?

Regression



We're predicting a continuous target variable. Supervised.

Recall Conditional Distributions

Our data has the form $(x_1, y_1), \dots, (x_N, y_N) \sim f(x, y)$.

For X and Y discrete, the distribution of Y given that we have observed $X = x$ is the *conditional probability function*

$$f(y|x) = \mathbb{P}(Y = y|X = x) = \frac{\mathbb{P}(X = x, Y = y)}{\mathbb{P}(X = x)} = \frac{f(x, y)}{f(x)} \quad (11)$$

and in general we have

$$f(x, y) = f(x|y)f(y) = f(y|x)f(x) \quad (12)$$

and Bayes' rule:

$$f(y|x) = \frac{f(x|y)f(y)}{\int f(x|y)f(y)dy}. \quad (13)$$

Regression

One way to summarize the relationship between the target Y and the features X is through the *regression function*

$$r(x) = \mathbb{E}(Y|X = x) = \int y f(y|x) dy, \quad (14)$$

which gives the single best guess for y given a value of x . Keep in mind that this is less information than is contained in the entire conditional pdf – estimating the latter is a different problem, called *conditional density estimation*.

Perhaps the simplest useful regression method is *linear regression*, where $r(x)$ is assumed to be linear.

Linear Regression

We assume that $\mathbb{V}(Y|X = x) = \sigma^2$ does not depend on x :

$$y = r(x) + \epsilon = \beta_0 + \beta_1 x + \epsilon \quad (15)$$

where $\mathbb{E}(\epsilon|X) = 0$ and $\mathbb{V}(\epsilon|X) = \sigma^2$, in one dimension. β_1 is the slope and β_0 is the intercept.

Task: regression

Model class: all possible linear regressors (parametric)

Loss: squared error

Optimizer: linear algebra

Generalization mechanism: AIC/BIC (BIC to be discussed later)

Linear Regression

The general multivariate version is sometimes called *multiple linear regression*.

The data are pairs $(x_1, y_1), \dots, (x_N, y_N)$ and the weights are a vector β . To include an intercept term, we add a feature to the data which is all 1's, usually making it the first column. The model is then

$$y_i = \mathbf{X}_i \beta + \epsilon_i = \sum_j \beta_j \mathbf{X}_{ij} + \epsilon_i. \quad (16)$$

In this context we'll call the vector of target values \mathbf{Y} , the data matrix \mathbf{X} whose rows are the data vectors, and the vector of residuals ϵ .

Linear Regression

How do we minimize the mean squared error (MSE), or (unnormalized) the *residual sum of squares*?

$$\text{RSS}(\beta) = \sum_{i=1}^N \left(y_i - \hat{f}(x_i) \right)^2 \quad (17)$$

$$= (\mathbf{Y} - \mathbf{X}\beta)^T (\mathbf{Y} - \mathbf{X}\beta) \quad (18)$$

This is a quadratic function in the $D + 1$ parameters. We can differentiate with respect to β and set it to zero:

$$\frac{\delta \text{RSS}(\beta)}{\delta \beta} = -2\mathbf{X}^T (\mathbf{Y} - \mathbf{X}\beta) = 0. \quad (19)$$

Linear Regression

Assuming that the $(D + 1) \times (D + 1)$ matrix $\mathbf{X}^T \mathbf{X}$ is invertible,

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} \quad (20)$$

$$\mathbb{V}(\hat{\beta}) = \sigma^2 (\mathbf{X}^T \mathbf{X})^{-1} \quad (21)$$

$$\hat{\beta} \xrightarrow{p} \beta \quad (22)$$

$$\hat{\beta} \approx N(\beta, \sigma^2 (\mathbf{X}^T \mathbf{X})^{-1}). \quad (23)$$

An unbiased estimator of σ^2 is

$$\hat{\sigma}^2 = \left(\frac{1}{N - (D + 1)} \right) \sum_{i=1}^N \hat{\epsilon}_i^2 \quad (24)$$

where $\epsilon_i = \hat{f}(x_i) - y_i$, called the *residual*.

Linear Regression

An approximate $1 - \alpha$ confidence interval for β_j is

$$\hat{\beta}_j \pm z_{\alpha/2} \widehat{\text{se}}(\hat{\beta}_j) \quad (25)$$

where $\widehat{\text{se}}^2(\hat{\beta}_j)$ is the j^{th} diagonal element of the matrix $\hat{\sigma}^2(\mathbf{X}^T \mathbf{X})^{-1}$.

We can also obtain an exact confidence interval, which involves a subtle correction.

Note: Linear regression with transformed features effectively becomes a more powerful model - sometimes this is called *generalized linear regression*.

Ridge Regression

Now consider a modification in which we now minimize

$$\sum_{i=1}^N \left(y_i - \beta_0 - \sum_{j=1}^D x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^D \beta_j^2 \quad (26)$$

or

$$\text{RSS}_{\text{ridge}}(\beta) = (\mathbf{Y} - \mathbf{X}\beta)^T (\mathbf{Y} - \mathbf{X}\beta) + \lambda \beta^T \beta \quad (27)$$

which is the original RSS plus a term which penalizes the size of the weights. λ is a complexity parameter which controls the amount of *regularization*, or *shrinkage*, or *weight decay* (in the neural net literature).

Ridge Regression

This is equivalent to minimizing

$$\sum_{i=1}^N \left(y_i - \beta_0 - \sum_{j=1}^D x_{ij} \beta_j \right)^2 \quad (28)$$

$$\text{subject to } \sum_{j=1}^D \beta_j^2 \leq s \quad (29)$$

where s now performs the function of λ .

The least-squares solution is

$$\hat{\beta}_{\text{ridge}} = (\mathbf{X}^T \mathbf{X} + \lambda I)^{-1} \mathbf{X}^T \mathbf{Y}. \quad (30)$$

Ridge Regression: Why?

Why do it?

- Indeterminacy (*e.g.* a large weight can be balanced by a negative weight on a correlated feature) adds variance.
- By having a complexity parameter, we have a way to account for test error, since we can cross-validate for λ .
- Some weights will go to zero, effectively making this a feature selection mechanism. This is good for interpretability, and possibly computation.

Ridge Regression: Why?

- Adding a constant to the diagonal before inversion justifies a numerical trick – making $\mathbf{X}^T \mathbf{X}$ non-singular even if it is not of full rank.
- Bayesian interpretation: Ridge regression can also be derived as the mean or mode of a posterior distribution if we assume $y_i \sim N(x_i^T \beta, \sigma^2)$ with each parameter β_j having prior distribution $N(0, \tau^2)$. Then the negative log-posterior density of β is exactly $\text{RSS}_{\text{ridge}}(\beta)$, with $\lambda = \sigma^2 / \tau^2$.

Lasso Regression

More practically interesting is *the lasso*, which minimizes

$$\sum_{i=1}^N \left(y_i - \beta_0 - \sum_{j=1}^D x_{ij} \beta_j \right)^2 \quad (31)$$

$$\text{subject to } \sum_{j=1}^D |\beta_j| \leq t. \quad (32)$$

This change in the constraint has the effect of making the zeroing of weights, or feature selection, more aggressive. Unfortunately it also makes the solutions nonlinear in y_i , making the optimization problem a *quadratic program*, a more difficult type of optimization problem which we will discuss in a later lecture.