Semi-supervised Principal Components Regression

Charles Zheng

October 25, 2016

Abstract

Semi-supervised learning refers to the problem of learning a rule for predicting labels y from features x, given training data which includes both labeled examples $(x_1, y_1), \ldots, (x_\ell, y_\ell)$ and unlabeled examples $x_{\ell+1}, \ldots, x_n$. A basic question of this field is to characterize the conditions under which the unlabeled examples can be used to improve generalization error. We introduce a latent variable model for which semi-supervised principal components regression is shown to outperform supervised ridge regression.

Keywords: Semi-supervised, ridge regression, principal components analysis, latent variables

1 Introduction

1.1 Ridge Regression

Ridge regression is a linear method for predicting a real-valued label $y \in \mathbb{R}$ from a vector of real-valued features $x \in \mathbb{R}^p$. It is produces a regularized least squares estimate for a coefficient vector $\hat{\beta}_\lambda$ and intercept $\hat{c}_\lambda$, where λ is a regularization parameter. This coefficient vector $\hat{\beta}_\lambda$ is used to predict the label y_* for an unlabeled example with features x_* by the rule

$$\hat{y}_\lambda = \hat{\beta}_\lambda^T x_* + \hat{c}_\lambda$$

Given training data with labels $Y = (y_1, \ldots, y_n)$ and features $X = (x_1^T, \ldots, x_n^T)$, and asssuming the feature matrix X is normalized to have columns with zero mean and unit variance, the ridge regression coefficient vector is defined as

$$\hat{\beta}_\lambda = (X^T X + \lambda I)^{-1} X^T Y_c$$

and the intercept as

$$\hat{c}_\lambda = \bar{y}$$

where $Y_c = (y_1 - \bar{y}, \dots, y_n - \bar{y})$ and $\bar{y} = \frac{1}{n} 1^T Y$.

The dependence of the ridge-regression prediction $\hat{y}_\lambda$ on the penalty λ can be easily seen by an application of the singular-value decomposition. Suppose for now that $n < p$. Write the singular-value decomposition of X as

$$\underbrace{X}_{n \times p} = \underbrace{U}_{n \times p} \underbrace{D}_{p \times p} \underbrace{V^T}_{p \times p}$$

Then it is evident that

$$\hat{\beta}_\lambda = VD(D^2 + \lambda)^{-1} U^T Y$$

and therefore, writing $z_* = D^{-1} V^T x_*$, and $D = \text{diag}(d_1, \dots, d_p)$,

$$\hat{y}_\lambda = z_*^T \text{diag}\left(\frac{d_i^2}{d_i^2 + \lambda}\right) U^T Y + \bar{y}$$

We see that increasing λ tends to shrink the prediction $\hat{y}$ towards $\bar{y}$. Specifically, ridge regression shrinks all of the principal directions $U^T Y$, applying more shrinkage to the directions of lowest variance d_i (Hastie 2008).

Shrinking $\hat{y}$ increases squared bias (defined as $\mathbf{E}_{X, x_*, y_*}(\mathbf{E}_Y[\hat{y}] - y_*)^2)$ but also reduces variance (defined as $\mathbf{E}_{X, x_*, y_*}[(\hat{y} - \mathbf{E}_Y[\hat{y}])^2]$). The optimal λ is determined by this bias-variance tradeoff, and in practice, can be estimated in a data-depndent way, e.g. by using cross-validation.

Due to the importance of choosing λ, it is usually not practical to study the properties of ridge regression for fixed λ. One way to approach for studying ridge regression is to let $f : X, Y \to \mathbb{R}$ be a selection rule for choosing λ in a data-dependent way, and define the ridge regression prediction rule with selection rule f by

$$\hat{y}_f = \hat{\beta}^T_{f(X,Y)} x_* + \hat{c}_{f(X,Y)}$$

Then the risk for the selection rule f is

$$R_f = \mathbf{E} ||\hat{y}_f - y_*||^2$$

However, most selection rules f used in practice are difficult to analyze. Therefore, a more tractable approach is to define an *oracle*-guided ridge regression procedure, which uses

$$\lambda_{oracle} = \text{argmin}_\lambda \mathbf{E}_{X, Y, x_*, y_*}[||y_* - \hat{y}_\lambda||^2]$$

and hence incurs the risk

$$R_{oracle} = \mathbf{E}||\hat{y}_{\lambda_{oracle}} - y_*||^2$$

While it might appear that R_{oracle} is obviously smaller than R_f for any selection rule f, since R_{oracle} uses information about the unknown join distribution of (x, y), in fact there is no existing proof that R_{oracle} is necessarily smaller than R_f. Nevertheless, it is widely believed that R_{oracle} should be close to R_f for selection rules f based on cross-validation.

1.2 Principal Components Regression

Principal components regression applies ordinary least-squares linear regression to the top k principal components of X. Recalling the singular-value decomposition $X = UDV^T$ from the previous section, let V_k denote the first K columns of V, and let $T_k = XV_k$. For a new point x_*, let $t_* = V_k^T x_*$ Define the k-principal components regression coefficient vector

$$\hat{\gamma}_{PCR-k} = (T_k^T T_k)^{-1} T_k^T Y$$

and the prediction rule as

$$\hat{y}_{PCR-k} = t_*^T \hat{\gamma}_{PCR-k} + \hat{c}_{PCR-k}$$

where if X is centered,

$$\hat{c} = \bar{y}$$

It is easy to compare principal components regression with ridge regression by writing $\hat{y}_{PCR-k}$ in terms of the SVD of X,

$$\hat{y}_{PCR-k} = z_*^T \mathrm{diag}(1_k, 0) U^T Y + \bar{y}$$

where as before $z^* = Vx_*$. While ridge regression shrinks the principal directions $U^T Y$ depending on the variance of the corresponding principal components, principal components regression “kills” all but the top k principal directions without shrinking the remaining k. Hence, in a supervised setting PCR and ridge regression have very similar behavior. However, we will see that unlike with ridge regression, PCR has the potential to improve with the addition of unlabeled examples.

2 Theory

2.1 Model

We specify a generative model for data matrices $X = (x_1^T, \dots, x_n^T)$ and $Y = (y_1, \dots, y_n)$, which together represent n labeled examples.

Let $Z = (z_1^T, \dots, z_n^T)$ be an $n \times r$ matrix of latent variables, where each entry z_{ij} is iid standard normal. The latent variables are related to X and Y in the following way. Let

$$X = Z\alpha + 1_n C^T + E$$

where α is a $r \times p$ coefficient matrix with unit-norm columns, C is a fixed $p \times 1$ intercept matrix, and E is a $n \times p$ random matrix of error terms which are iif $N(0, \sigma_\epsilon^2)$. Similarly, let

$$Y = Z\gamma + c + \epsilon$$

where γ is a $r \times 1$ coefficient vector, c is an intercept term, and ϵ is a $n \times 1$ vector with entries iid $N(0, \sigma_\epsilon^2)$.

The semi-supervised learning problem can be posed as follows. Let $\ell < n$ be the number of labeled examples, and let Y_ℓ be the first ℓ rows of Y. Supposing that only X and Y_ℓ are observed, and α, γ, and σ_ϵ^2 are unknown parameters, the problem is to predict $\hat{Y}$ for all n examples. The squared-error loss is defined as

$$||\hat{Y} - Y||^2$$

and the goal is to choose a prediction method which minimizes the *risk*, or expected squared-error loss.

Of course, one can always set the first ℓ entries of $\hat{Y}$ to be equal to Y_ℓ, which guarantees zero error on those examples. Thus the challenge is to make a prediction for the unobserved entries of Y.

Throughout the paper we make the simplifying assumption that r, the number of latent variables, is known; however, the problem of testing the number of principal components r is well-studied in the statistics literature, and our analysis could be extended to incorporate the case of unknown r. Noting that the problems of estimating the unknown intercept terms C and c as well as the marginal variances of X and Y are also well-understood, we lose little theoretical power and gain much clarity by making additional assumptions that $C = 0$, $c = 0$, and that all the columns of α and γ are unit-norm. Note that as a result, X and Y have zero marginal mean and equal marginal variances of $1 + \sigma_\epsilon^2$, justifying the omission of the normalization step normally employed in ridge regression.

2.2 Ridge Regression

Write the following singular value decompositions

$$\underbrace{\alpha}_{r\times p} = \underbrace{\eta}_{r\times p} \underbrace{\tilde{D}}_{p\times p} \underbrace{\tilde{V}^T}_{p\times p}$$

$$\underbrace{X}_{r\times p} = \sqrt{n} \underbrace{U}_{n\times p} \underbrace{D}_{p\times p} \underbrace{V^T}_{p\times p}$$

The scaling of $\sqrt{n}$ is so that D is $O(1)$. Also rescale the ridge regression penalty so that

$$\hat{\beta}_\lambda = (X^T X + n\lambda I)^{-1} X^T Y$$

Write

$$\begin{aligned} Z^T Z &= n(I + \delta_Z) \\ \tilde{V}^T V &= I + \delta_V \end{aligned}$$

Then

$$\begin{aligned} \hat{y}_\lambda - y_* =& x_*^T (X^T X + n\lambda I)^{-1} X^T Y - y_* \\ =& n^{-1}(z_*^T \alpha + E_*) V (D^2 + \lambda I)^{-1} V^T (\alpha^T Z^T Z\gamma + E^T Z\gamma + \alpha^T Z^T \epsilon + E^T \epsilon) - z_*^T \gamma - \epsilon \\ =& z_*^T (\eta \tilde{D}^2 (D^2 + \lambda)^{-1} \eta^T \gamma - \gamma) \\ &+ z_*^T (\eta \tilde{D} \delta_v (D^2 + \lambda)^{-1} (I + \delta_V^T) \tilde{D}^T \eta^T (I + \delta_Z)\gamma) \\ &+ z_*^T (\eta \tilde{D} (I + \delta_V)(D^2 + \lambda)^{-1} \delta_V^T \tilde{D}^T \eta^T (I + \delta_Z)\gamma) \\ &+ z_*^T (\eta \tilde{D} (I + \delta_V)(D^2 + \lambda)^{-1} (I + \delta_V^T) \tilde{D}^T \eta^T \delta_Z \gamma) \\ &+ E_* V (D^2 + \lambda I)^{-1} V^T \alpha^T (I + \delta_Z)\gamma \\ &+ n^{-1} x_*^T V (D^2 + \lambda I)^{-1} V^T (E^T Z\gamma + \alpha^T Z^T \epsilon + E^T \epsilon) - \epsilon \end{aligned}$$

3 References

- Hastie, Tibshirani, Friedman. “The Elements of Stastistical Learning,” 2008.

- Zhu, X. “Semi-supervised learning literature survey.” 2005.

- Niyogi, P. “Manifold Regularization and semi-supervised learning: Some theoretical analysis.” 2008.