Sharif University of Technology
Department of Electrical Engineering

Deep Learning HW1 solutions

Instructor: Dr. E. Fatemizadeh Fall Semester 1403

SVM for Classification

1

In the linearly separable case, if one of the training samples is removed:

1. If the point is not a support vector, then the margin remains unchanged.
2. If the point is a support vector, then the margin length can become larger and move towards the point which is removed if the point was the only support vector, or remain unchanged otherwise.

Logistic regression focuses on maximizing the probability of the data. The farther the data lies from the separating hyperplane, the more it favors logistic regression (LR) as opposed to SVM, which tries to explicitly find the maximum margin. If a point is not a support vector, it does not really affect the SVM margin. Since LR is a density estimation technique, each point will carry some weight and have some effect on the decision boundary.

2

The hinge loss is the upper bound on the number of misclassified instances. Here, we choose the hinge loss function as:

$$h(z) = \max(0, 1 - z).$$

The primal optimization of SVM is given by:

$$\min_{w,\xi_i} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i$$

where the slack variable ξ_i appears in one constraint, and we try to minimize it. This leads to the constraint:

$$y_i(w^T x_i) \geq 1 - \xi_i \quad \text{or} \quad \xi_i \geq 0.$$

The slack variable ξ_i is the tighter/larger one of the two values. So it can either be zero or $1 - y_i(w^T x_i)$. Thus,

$$\xi_i = \max(0, 1 - y_i(w^T x_i)),$$

which is the hinge loss function. Therefore,

$$\xi_i = \max(0, 1 - y_i(w^T x_i)) = h(y_i(w^T x_i)).$$

We know that the hinge loss is the upper bound on the number of misclassified instances, so the upper bound is given by:

$$\sum_{i=1}^{n} h(y_i(w^T x_i)) \quad \text{or simply} \quad \sum_{i=1}^{n} (\xi_i > 1).$$

3

C is the trade-off parameter that determines whether we prioritize a small norm of w (indicating a large margin) or no violations of the margin constraints (indicating a small sum of hinge loss).

- When $C \to 0$, more emphasis is given to finding the largest margin, allowing for some noise without penalizing misclassification.
- When $C \to \infty$, we place higher weight on margin constraint violations, leading to a hyperplane where the required slack is minimized, even at the expense of the margin.

4

When two classes are linearly separable, both Hard SVM and Logistic Regression can find a solution. The major difference is that Logistic Regression finds a decision boundary that maximizes its likelihood function, while Hard SVM finds a decision boundary with maximal margin.

5

When the two classes are not linearly separable, Logistic Regression will still find a decision boundary that maximizes its likelihood function. For Soft SVM, it finds a decision boundary that best balances the margin and errors. The key difference between SVM and Logistic Regression is that SVM is a geometry-motivated model, while Logistic Regression is a probability-motivated model.

PCA

1.1

This is simply a change of basis. It follows from:

$$(\hat{x}_i - \hat{x}_j)^T(\hat{x}_i - \hat{x}_j) = (z_i - z_j)^T V_{1:k}^T V_{1:k}(z_i - z_j) = (z_i - z_j)^T(z_i - z_j),$$

since the columns of $V_{1:k}$ are orthogonal.

1.2

Suppose V is the full $p \times p$ matrix containing all p eigenvectors. Let $\tilde{V} = V_{k+1:p}$, the matrix consisting of all eigenvectors except the first k. Since V is orthogonal, we have:

$$VV^T = I = V_{1:k}V_{1:k}^T + \tilde{V}\tilde{V}^T.$$

The reconstruction error is given by:

$$\sum_{i=1}^{n} \|x_i - \hat{x}_i\|_2^2 = \sum_{i=1}^{n} \|x_i - V_{1:k}V_{1:k}^T x_i\|_2^2 = \sum_{i=1}^{n} \|(I - V_{1:k}V_{1:k}^T)x_i\|_2^2.$$

Expanding this, we get:

$$= \sum_{i=1}^{n} \|\tilde{V}\tilde{V}^T x_i\|_2^2 = \sum_{i=1}^{n} x_i^T \tilde{V}\tilde{V}^T\tilde{V}\tilde{V}^T x_i = \sum_{i=1}^{n} x_i^T \tilde{V}\tilde{V}^T x_i.$$

Thus,

$$= \sum_{i=1}^{n} \text{Tr}(x_i^T \tilde{V} \tilde{V}^T x_i) = \sum_{i=1}^{n} \text{Tr}(\tilde{V}^T x_i x_i^T \tilde{V}) = (n-1)\,\text{Tr}(\tilde{V}^T S \tilde{V}),$$

where S is the sample covariance matrix. Finally, we have:

$$= (n-1) \sum_{j=k+1}^{p} \lambda_j,$$

where in the last step we used $S\tilde{V} = \tilde{V}\Lambda_{k+1:p}$ with $\Lambda_{k+1:p} = \text{diag}(\lambda_{k+1}, \lambda_{k+2}, \ldots, \lambda_p)$.

3.

Consider the equation $Xw = y$, where $X \in \mathbb{R}^{m \times n}$ is a non-square data matrix, w is a weight vector, and y is a vector of labels corresponding to the datapoints in each row of X.
Let's say that $X = U\Sigma V^T$ is the (full) SVD of X. Here, U and V are orthonormal square matrices, and Σ is an $m \times n$ matrix with non-zero singular values (σ_i) on the "diagonal".
For this problem, we define Σ^+ as an $n \times m$ matrix with the reciprocals of the singular values $\left(\frac{1}{\sigma_i}\right)$ along the "diagonal".

(a) How do we find the weights w that minimizes the error between Xw and y?

First, consider the case where $m > n$, i.e. our data matrix X has more rows than columns (tall matrix) and the system is overdetermined. **How do we find the weights w that minimizes the error between Xw and y?** In other words, we want to solve:

$$\min_w \|Xw - y\|^2.$$

Solution: It is FULL SVD, U and V are square orthonormal matrices.
This is the classic least squares problem. The solution is given by

$$\hat{w} = (X^T X)^{-1} X^T y.$$

This can be derived from vector calculus, and also has an elegant interpretation in the context of orthogonal projection of y on the column space of X.

(b) Plug in the SVD $X = U\Sigma V^T$ and simplify.

Solution:

$$(X^T X)^{-1} X^T = (V\Sigma^T U^T U \Sigma V^T)^{-1} V \Sigma^T U^T.$$

Since U has orthonormal columns, $U^T U = I$. Notice $\Sigma^T \Sigma$ is a square, $n \times n$ diagonal matrix with squared singular values σ_i^2 along the diagonal.

$$(X^T X)^{-1} X^T = (V\Sigma^T \Sigma V^T)^{-1} V \Sigma^T U^T.$$

Apply the fact that $(AB)^{-1} = B^{-1}A^{-1}$, and that $V^{-1} = V^T$ since the matrix is orthonormal.

$$(X^T X)^{-1} X^T = V(\Sigma^T \Sigma)^{-1} V^T V \Sigma^T U^T.$$

Simplify since $V^T V = I$.

$$(X^T X)^{-1} X^T = V(\Sigma^T \Sigma)^{-1} \Sigma^T U^T.$$

Notice that $(\Sigma^T\Sigma)^{-1}\Sigma^T$ is an $n \times m$ matrix with the reciprocals of the singular values $\frac{1}{\sigma_i}$ on the "diagonal". We can call this matrix Σ^+. Note that this isn't a true matrix inverse (since the matrix Σ is not square). So we can write our answer as

$$(X^TX)^{-1}X^T = V\Sigma^+U^T.$$

You should draw out the matrix shapes and convince yourself that all the matrix multiplications make sense.

(c) What happens if we left-multiply X by our matrix A?

You'll notice that the least-squares solution is in the form $w^* = Ay$. **What happens if we left-multiply X by our matrix A?** This is why the matrix A of the least-squares solution is called the left-inverse.

Solution:

$$(X^TX)^{-1}X^TX = I.$$

We can also see this from our SVD interpretation.

$$V\Sigma^+U^TU\Sigma V^T = V\Sigma^+\Sigma V^T = VV^T = I.$$

This is why the least-squares solution is called the left-inverse.

(d) What is the minimum norm solution?

Now, let's consider the case where $m < n$, i.e. the data matrix X has more columns than rows and the system is underdetermined. There exist infinitely many solutions for w, but we seek the minimum-norm solution, i.e., we want to solve $\min \|w\|$ s.t. $Xw = y$. **What is the minimum norm solution?**

Solution: The min-norm problem is solved by

$$w = X^T(XX^T)^{-1}y.$$

We can see this by choosing w that has a zero component in the nullspace of X, and thus w is in the range of X^T. Alternatively, one can write the Lagrangian, take the dual, apply the KKT conditions, and solve to get the same answer.

(e) Plug in the SVD $X = U\Sigma V^T$ and simplify.

Solution:

$$X^T(XX^T)^{-1} = (U\Sigma V^T)(U\Sigma V^T)^T(U\Sigma V^T)^{-1} = V\Sigma^TU^TU\Sigma^TU^T(V^T)^{-1}$$

$$= V\Sigma^TUU^TU(\Sigma^T\Sigma)^{-1}U^T = V\Sigma^T(\Sigma^T\Sigma)^{-1}U^T.$$

Here, we have that $\Sigma^T(\Sigma^T\Sigma)^{-1}$ is an $n \times m$ matrix with the reciprocals of the singular values, $\frac{1}{\sigma_i}$, on the "diagonal". We can call this matrix Σ^+ so that we have

$$= V\Sigma^+U^T.$$

(f) What happens if we right-multiply X by our matrix B?

You'll notice that the min-norm solution is in the form $w^* = By$. **What happens if we right-multiply X by our matrix B?** This is why the matrix B of the min-norm solution is called the right-inverse.
Solution: Similar to the previous part,

$$XX^T(XX^T)^{-1} = I.$$

This can also be seen from the SVD perspective. This is why the min-norm solution is called the right-inverse.

4.

Consider a linear regression problem with n training points and d features. When $n = d$, the feature matrix $F \in \mathbb{R}^{n\times n}$ has some maximum singular value α and an extremely tiny minimum singular value. We have noisy observations $y = Fw^* + \epsilon$. If we compute $w_{inv} = F^{-1}y$, then due to the tiny singular value of F and the presence of noise, we observe that $\|w_{inv} - w^*\|_2 = 10^{10}$. Suppose instead of inverting the matrix we decide to use gradient descent instead. We run k iterations of gradient descent to minimize the loss $\ell(w) = \frac{1}{2}\|y - Fw\|^2$ starting from $w_0 = 0$. We use a learning rate η which is small enough that gradient descent cannot possibly diverge for the given problem. (This is important. You will need to use this.)
The gradient-descent update for $t > 0$ is:

$$w_t = w_{t-1} - \eta\left(F^T(Fw_{t-1} - y)\right).$$

We are interested in the error $\|w_k - w^*\|_2^2$. We want to show that in the worst case, this error can grow at most linearly with iterations k and in particular $\|w_k - w^*\|_2 \le k\eta\|y\|_2 + \|w^*\|_2$. i.e. the error cannot go "nuts," at least not very fast.
For the purposes of the homework, you only have to prove the key idea, since the rest follows by applying induction and the triangle inequality.
Show that for $t > 0$, $\|w_t\|_2 \le \|w_{t-1}\|_2 + \eta\alpha\|y\|_2$.
(HINT: What do you know about $(I - \eta F^T F)$ if gradient descent cannot diverge? What are its eigenvalues like? Use this fact.)

Solution: We have,

$$\begin{aligned}
\|w_t\|_2 &= \|w_{t-1} - \eta(F^T(Fw_{t-1} - y))\|_2 \\
&= \|(I - \eta F^T F)w_{t-1} + \eta F^T y\|_2 \\
&\le \|(I - \eta F^T F)w_{t-1}\|_2 + \eta\|F^T y\|_2 \\
&\le \sigma_{\max}(I - \eta F^T F)\|w_{t-1}\|_2 + \eta\sigma_{\max}(F)\|y\|_2 \\
&\le \|w_{t-1}\|_2 + \eta\alpha\|y\|_2,
\end{aligned}$$

where we used the fact that η is chosen so that gradient descent does not diverge so the maximum eigenvalue of $(I - \eta F^T F)$ cannot have an absolute value greater than 1. But $I - F^T F$ is a real symmetric matrix and the spectral theorem tells us that the singular values are therefore just the absolute values of the eigenvalues. This means that the maximum singular value also must have absolute value less than 1.

5.

(a) Show that we can decompose the expected mean squared error into three parts: bias, variance, and irreducible error σ^2.

Formally, suppose we have a randomly sampled training set $\mathcal{D}$ (drawn independently of our test data), and we select an estimator denoted $\hat{\theta} = \hat{\theta}(\mathcal{D})$ (for example, via empirical risk minimization). The expected mean squared error for a test input x can be decomposed as below:

$$\mathbb{E}_{Y\sim p(y|x),\mathcal{D}}\left[(Y - f_{\hat{\theta}(\mathcal{D})}(x))^2\right] = \text{Bias}(f_{\hat{\theta}(\mathcal{D})}(x))^2 + \text{Var}(f_{\hat{\theta}(\mathcal{D})}(x)) + \sigma^2$$

You may find it helpful to recall the formulaic definitions of Variance and Bias, reproduced for you below:

$$\text{Bias}(f_{\hat{\theta}(\mathcal{D})}(x)) = \mathbb{E}_{Y\sim p(y|x),\mathcal{D}}\left[f_{\hat{\theta}(\mathcal{D})}(x) - Y\right]$$

$$\text{Var}(f_{\hat{\theta}(\mathcal{D})}(x)) = \mathbb{E}_{\mathcal{D}}\left[(f_{\hat{\theta}(\mathcal{D})}(x) - \mathbb{E}[f_{\hat{\theta}(\mathcal{D})}(x)])^2\right]$$

Solution: For simplicity of notation, let $\mathbb{E}[\cdot]$ denote $\mathbb{E}_{Y\sim p(y|x),\mathcal{D}}[\cdot]$.

$$\begin{aligned}\mathbb{E}\left[(Y - f_{\hat{\theta}(\mathcal{D})}(x))^2\right] &= \mathbb{E}\left[(Y - f_{\hat{\theta}(\mathcal{D})}(x))^2\right] \\ &= \mathbb{E}[f_{\hat{\theta}(\mathcal{D})}(x)^2 - 2Y f_{\hat{\theta}(\mathcal{D})}(x) + Y^2]\end{aligned}$$

By independence of Y and $\mathcal{D}$ and linearity of expectation,

$$\mathbb{E}[(Y - f_{\hat{\theta}(\mathcal{D})}(x))^2] = \mathbb{E}[f_{\hat{\theta}(\mathcal{D})}(x)^2] - 2\mathbb{E}[Y]\mathbb{E}[f_{\hat{\theta}(\mathcal{D})}(x)] + \mathbb{E}[Y^2]$$

Noting the definition of variance,

$$\begin{aligned}\mathbb{E}[(Y - f_{\hat{\theta}(\mathcal{D})}(x))^2] &= \text{Var}(f_{\hat{\theta}(\mathcal{D})}(x)) + \mathbb{E}[f_{\hat{\theta}(\mathcal{D})}(x)]^2 - 2\mathbb{E}[Y]\mathbb{E}[f_{\hat{\theta}(\mathcal{D})}(x)] + \mathbb{E}[Y^2] \\ &= \text{Var}(f_{\hat{\theta}(\mathcal{D})}(x)) + (\mathbb{E}[f_{\hat{\theta}(\mathcal{D})}(x)] - \mathbb{E}[Y])^2 + \text{Var}(Y|X = x) \\ &= \text{Var}(f_{\hat{\theta}(\mathcal{D})}(x)) + \text{Bias}(f_{\hat{\theta}(\mathcal{D})}(x))^2 + \text{Var}(Y|X = x)\end{aligned}$$

The conditional variance $\text{Var}(Y|x)$, which we will denote σ^2, captures the irreducible error that will be incurred no matter what learner $\hat{\theta}$ we use.

(b) Compute the Bias and Covariance of $\hat{\theta}$

Suppose our training dataset consists of $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, where the only randomness is coming from the label vector $Y = X\theta^* + \epsilon$, where θ^* is the true underlying linear model and each noise variable ϵ_i is i.i.d. with zero mean and variance 1. We use ordinary least squares to estimate a $\hat{\theta}$ from this data. **Calculate the bias and covariance of the $\hat{\theta}$ estimate and use that to compute the bias and variance of the prediction at particular test inputs x.**
Recall that the OLS solution is given by:

$$\hat{\theta} = (X^T X)^{-1} X^T Y,$$

where $X \in \mathbb{R}^{n \times d}$ is our (nonrandom) data matrix, $Y \in \mathbb{R}^n$ is the (random) vector of training targets. For simplicity, assume that $X^T X$ is diagonal.

Solution: We first compute the bias of $\hat{\theta}$. Recalling that we have $Y = X\theta^* + \epsilon$ for a noise vector ϵ, we then have:

$$\begin{aligned}
\mathbb{E}[\hat{\theta}] &= \mathbb{E}[(X^T X)^{-1} X^T (X\theta^* + \epsilon)] \\
&= \mathbb{E}[\theta^* + (X^T X)^{-1} X^T \epsilon] \\
&= \theta^* + (X^T X)^{-1} X^T \mathbb{E}[\epsilon] \\
&= \theta^* \quad \text{(since } \epsilon \text{ has 0 mean).}
\end{aligned}$$

So our bias is:

$$\text{bias} = \mathbb{E}[\hat{\theta} - \theta] = \mathbb{E}[\hat{\theta}] - \theta^* = 0.$$

We thus see that the OLS estimator $\hat{\theta}$ is an **unbiased** estimator of the true parameter θ.

Next, we compute the variance of $\hat{\theta}$, and we will proceed by first computing the covariance of $\hat{\theta}$:

$$\begin{aligned}
\mathbb{E}[(\hat{\theta} - \theta^*)(\hat{\theta} - \theta^*)^T] &= \mathbb{E}[(X^T X)^{-1} X^T \epsilon \epsilon^T X (X^T X)^{-1}]. \\
&= (X^T X)^{-1} X^T \mathbb{E}[\epsilon \epsilon^T] X (X^T X)^{-1} \\
&= (X^T X)^{-1} X^T I_n X (X^T X)^{-1} \\
&= (X^T X)^{-1} X^T X (X^T X)^{-1} \\
&= (X^T X)^{-1}.
\end{aligned}$$

Now for a particular test input x, we can compute the variance:

$$\begin{aligned}
\text{Var}[x^T \hat{\theta}] &= \text{Var}[x^T (\hat{\theta} - \theta)] \\
&= \mathbb{E}[x^T (\hat{\theta} - \theta)(\hat{\theta} - \theta)^T x] \\
&= x^T (X^T X)^{-1} x.
\end{aligned}$$

For simplicity, suppose $X^T X$ were a diagonal matrix (we could have applied an orthogonal transformation to achieve this) with sorted entries $\sigma_1^2 \geq \sigma_2^2 \geq \cdots \geq \sigma_d^2$ (corresponding to the data variances in each dimension). Now we can easily compute the variance as $\sum_{i=1}^d \frac{x_i^2}{\sigma_i^2}$, and we see that in directions where σ_i is close to 0 (which means there is very little variance in the data in this dimension), the variance of our estimate can explode (and thus our risk as well).