Mathematical Introduction to Machine Learning

Lecture 3: Classification Problems

November 17, 2024

Lecturer: Lei Wu *Scribe: Lei Wu*

Note that when deriving the kernel version of SVM, the convention way is using the Lagrange duality and KKT condition. However, in this note, we instead apply the representer theorem to derive it and thus we avoid the introduction of Lagrange duality and KKT condition. By this derivation, the only minor issue is that it is unclear why the support vectors correspond to the points with the smallest margins. This issue, however, can be resolved easily by using the definition of minima (which is left to homework). If you would like to know more about Lagrange duality, we refer to Section 15 of tripleS's book.

1 Problem setup

- In classification problem, we are given n samples $(x_1, y_1), \dots, (x_n, y_n)$ with $y_i \in \{1, 2, \dots, C\}$. C is the number of classes. Different from regression, the labels are discrete.
- Our ultimate goal is to learn a classifier $f : \mathbb{R}^d \mapsto \{1, \dots, C\}$ to minimize the population *classiciation error*

  $$\mathcal{R}(f) = \mathbb{E}_{x,y}[\ell_{0-1}(f(x), y)],$$

  where

  $$\ell_{0-1}(y, y') = \begin{cases} 0 & \text{if } y = y' \\ 1 & \text{otherwise.} \end{cases}$$

- In practice, we can only deal with the empirical error

  $$\widehat{\mathcal{R}}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell_{0-1}(f(x_i), y_i) + \lambda\Omega(f), \tag{1}$$

  where the regularization function $\Omega : \mathcal{F} \mapsto \mathbb{R}^+$ implements a way of penalizing the model complexity and $\mathcal{F}$ is the hypothesis class.
- There is a serious issue with the optimization problem (1): the loss function ℓ_{0-1} is not continuous. Consequently, it is unlikely to directly optimize (1).

The core problem in classification is to design an appropriate "surrogate loss" to replace ℓ_{0-1}.

Here the "appropriate" means: 1) the corresponding optimization problem is easy to solve; 2) The minimization of surrogate loss can lead to small ℓ_{0-1} loss.

Unfortunately, there is no unifying approach to designing surrogate loss. In this lecture, we introduce the two most popular ones:

- probabilistic modeling: the softmax classifier,
- geometric modeling: the max-margin classifier.

2 The probabilistic modeling and softmax classifier

In many cases, we can model the label as a discrete probability distribution over the C classes. This typically happens when the data of different classes are not well separated. See the left panel of Figure 1 for an illustration. In such a case, probabilistic modelling of labels are better for data points that are close to the boundary of two classes. This situation also occur frequently in practice. See the right panel of Figure 1, which shows an image which contains both 'dog' and 'car' two classes, for which modeling the label as a probability distribution is obviously a better choice.

Figure 1: **Left:** An illustration of how two classes are not well separated. **Right:** An image of containing both dog and car, whose label should be modeled as a probability distribution over classes.

In probabilistic modeling of classification, the "true label" of x is a discrete prob. distribution:

$$y = p(x) = (p_1(x), p_2(x), \ldots, p_C(x)),$$

where for any $j \in [C]$ $p_j(x) \geq 0$ and $\sum_j p_j(x) = 1$. Denote by

$$\mathcal{P}_C = \{p \in \mathbb{R}^C : \sum_j p_j = 1, \min_j p_j \geq 0\}$$

the set of discrete probability distributions over C classes.

However, in practice, the "observed label" $y \in \{1, 2, \ldots, C\}$ is a sample drawn from the multinomial distribution parameterized by $p(x)$. An important related concept is the "one-hot label": $e_y = (0, \ldots, 0, 1, 0, \ldots, 0) \in \mathcal{P}_C$ where the nonzero coordinate is the y-th one. This can be viewed as a discrete "delta" probability distribution, which is a good inference of the true label $p(x)$ from just one sample.

Any C-dimensional vector can be converted to a discrete probability distribution by using the *softmax operator*: $\mathbb{R}^C \mapsto \mathcal{P}_C$,

$$\text{softmax}(z) = \left(\frac{e^{z_1/T}}{\sum_j e^{z_j/T}}, \frac{e^{z_2/T}}{\sum_j e^{z_j/T}}, \ldots, \frac{e^{z_C/T}}{\sum_j e^{z_j/T}} \right),$$

where $T > 0$ isa hyper-parameter, commonly referred to as the "temperature", that controls the "softness" of the softmax function. Given any parametrized model $f_\theta : \mathbb{R}^d \mapsto \mathbb{R}^C$, we can form a classifier by

$$F(x; \theta) = \text{softmax}(f_\theta(x)) : \mathbb{R}^d \mapsto \mathcal{P}_C.$$

Learning a classifier can formulated as following problem

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} d(F(x_i;\theta), e_{y_i}) + \lambda\Omega(\theta), \tag{2}$$

where $e_{y_i} \in \mathcal{P}_c$ is the one-hot label. What remains is to choose an appropriate "distance" function $\ell : \mathcal{P}_C \times \mathcal{P}_C \mapsto \mathbb{R}_{\geq 0}$ to measure the diffence between two probability distributions. There are a lot choices:

- $\|p-q\|_1 = \sum_j |p_j - q_j|$. This ℓ_1 distance corresponds to the TV norm.
- $\|p-q\|_2 = \sqrt{\sum_j |p_j - q_j|^2}$. This is the standard ℓ_2 norm.

However, the aforementioned choices do not take advantage of the unique structure of the output space, which is probabilistic in nature. The most commonly employed metric in practice is the Kullback-Leibler (KL) divergence

$$D_{KL}(p||q) = \sum_j p_j \log(p_j/q_j).$$

2.1 The KL divergence

The KL divergence, also known as *relative entropy*, is very popular in measuring the difference between two probability distributions. It is widely used in many problems such as the convergence analysis of MCMC sampler/Fokker-Plank equation.

For two probability distributions P, Q,

$$D_{KL}(P||Q) = \int \log(\frac{\mathrm{d}P}{\mathrm{d}Q})\,\mathrm{d}P,$$

where $\frac{\mathrm{d}P}{\mathrm{d}Q}$ is the Radon-Nikodynm derivative of P with respect to Q.

- When P, Q are discrete, $D_{KL}(P||Q) = \sum_j p_j \log(p_j/q_j)$.
- If P, Q have density function p, q, respectively, then $D_{KL}(P||Q) = \int p(x)\log(p(x)/q(x))\,\mathrm{d}x$.

Some basic facts of KL divergence:

- KL divergence is not symmetric, i.e., $D_{KL}(P||Q) \neq D_{KL}(Q||P)$;
- For any P, Q, $D_{KL}(P||Q) \geq 0$;
- $D_{KL}(P||Q) = 0$ iff $P = Q$ almost surely;
- D_{KL} is jointly convex, i.e.,

$$D_{KL}(\lambda P_1 + (1-\lambda)P_2||\lambda Q_1 + (1-\lambda)Q_2) \leq \lambda D_{KL}(P_1||Q_1) + (1-\lambda)D_{KL}(P_2||Q_2).$$

- $D_{KL}(P||Q) = -\sum_j p_j \log q_j + \sum_j p_j \log p_j = H(P,Q) - H(P)$. Here, $H(P,Q)$ denotes the **cross-entropy** between P and Q; $H(P)$ denotes the entropy of P.

2.2 Softmax classifier

For simplicity, we first consider one sample (x, y). Here, let $y \in \mathcal{P}_C$ be a general label (which is not necessarily one-hot).

$$D_{KL}(y||F(x;\theta)) = \sum_j y_j \log(\frac{y_j}{F_j(x;\theta)}) = \underbrace{C}_{\text{entropy of } y} \underbrace{-\sum_j y_j \log(F_j(x;\theta))}_{\text{cross-entropy}}, \tag{3}$$

where C is a constant independent of θ. Then minimizing the KL divergence is equivalent to minimizing the cross-entropy loss:

$$\min_\theta \frac{1}{n} \sum_{i=1}^n \left(-\sum_j y_{i,j} \log(F_{y_{i,j}}(x_i;\theta)) \right), \tag{4}$$

where $y_i = (y_{i,1}, \ldots, y_{i,C})$ corresponds to the label of i-th sample. When the label is one-hot, the objective can be further simplified to

$$\min_\theta -\frac{1}{n} \sum_{i=1}^n \log(F_{y_i}(x_i;\theta)), \tag{5}$$

where we abuse the notation of y_i for simplicity.

Remark 2.1. Can we minimize $D_{KL}(F(x;\theta)||y)$? Yes, we can. But the problem is that the resulting objective function is much more complicated.

Remark 2.2. In fact, the objective (5) can be easily derived from the maximum likelihood estimation (MLE) perspective. But the interpretation of minizing the KL divergence is more general and flexible as it precisely captures the structure of the label space. This viewpoint allows for straightforward customization of the classifier.

- Can we replace the softmax with other operators that map a vector to a discrete probability, e.g.,

$$s(z) = \left(\frac{|z_1|^\alpha}{\sum_j |z_j|^\alpha}, \ldots, \frac{|z_C|^\alpha}{\sum_j |z_j|^\alpha} \right)?$$

  The answer is affirmative. But it seems that the softmax performs better for many practical problems. The other choices might be useful in some specific problems.

- Can we replace the KL divergence with other distance functions? Yes!

- Can we encode the prior knowledge of the output probabilities? Yes! Given below is an example.

Entropy regularization. If our classification problem is fraught with considerable uncertainty or the observed labels are noisy, then fitting the one-hot labels would be a bad idea. We anticipate the output probability distribution to be smoother and exibit higher entropy, rather than behaving like a delta function. To address this issue, a direct approach is to encourage the entropy of output distribution:

$$\min_\theta \frac{1}{n} \sum_{i=1}^n D_{KL}(y_i||F(x_i;\theta)) - \lambda \,\text{entropy}(F(x_i;\theta)),$$

Label smoothing. Calculating entropy is often computationally intensive and costly. A more practical approach is to fit the *smoothed labels*:

$$S_\varepsilon : (0, \dots, 0, 1, 0, \dots, 0) \mapsto (\varepsilon, \dots, \varepsilon, 1 - (C-1)\varepsilon, \varepsilon, \dots, \varepsilon),$$

where ε should be sufficiently small to ensure that the smoothed label still represents a valid probability distribution. Utilizing these smoothed labels, which possess higher entropy, enables the development of a classifier with correspondingly higher entropy.

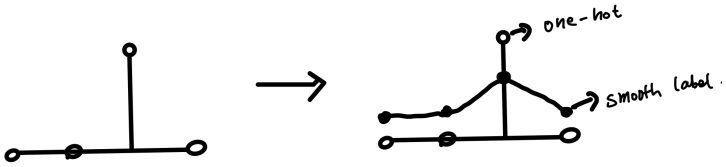

Figure 2: An illustration of label smoothing.

3 The SVM and max-margin classifier

When the data are well separated, we have that the predictions are very confident and output probability distributions are always like delta functions. In such a case, the probabilistic modeling of classification is not necessary since there are no uncertainty in prediction. Support vector machine (SVM) provides a geometric modeling of classification in such a case, and in particular, the introduced concept "margin" is very important for understanding classification. Throught this section, we will only consider the binary classification problem with the label $y \in \{-1, 1\}$ for simplicity.

3.1 Hard-SVM

Suppose that the data are *linearly separable* and hard-SVM attempt to find a classifier that separates the data with the largest "margin". See Figure 3 for an illustration. The basic idea is that we expect the decision boundary to be away from data points as far as possible. This intuition is formalized by using the concept of margin.

Figure 3 provides an illustration of hard SVM and as a comparison, a small-margin decision boundary is also ploted. In particular, the right panel provides a semantic visualization of how different classifiers perform on test data. One can see that the max-margin classifier performs much better than the small-margin classifier. The reason is that a classifier often makes mistakes only on data points that are close to decision boundary. For a max-margin classifier, the data that are close to decision boundary only contribute a small portion of whole population. In contrast, for a small-margin classifier, there are a large amount of points that are close to the decision boundary.

Next, we turn to provide a mathematical formulation/modeling of the above intuition/insight. Let $f_\theta(x) = \beta^T x + \beta_0$ with $\theta = (\beta, \beta_0)$ be the decision function. The corresponding decision boundary is given by the hyperplane:

$$H_\theta = \{x \in \mathbb{R}^d : \beta^T x + \beta_0 = 0\}.$$

Geometrically, the **margin** of sample x is defined as the distance between x to the decision boundary H_θ:

$$\gamma(x; \theta) = \text{dist}(x, H_\theta) = \frac{|\beta^T x + \beta_0|}{\|\beta\|}. \tag{6}$$

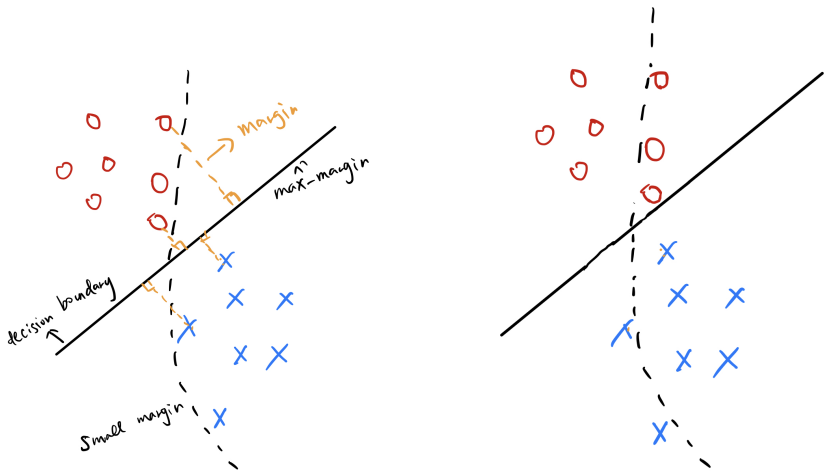

Figure 3: An illustration of hard-SVM on the training data (left) and test data (right). In particular, the right panel shows how the max-margin classifier performs much better than the small-margin classifier since the former are much more robust to the deviation between training and testing data.

For each sample, we can compute a margin. Thus, we have n margins:

$$\gamma(x_1;\theta), \gamma(x_2;\theta), \cdots, \gamma(x_n;\theta).$$

The next question is: what do we mean by maximizing the margin? Should we minimize the average margin $\frac{1}{n}\sum_{i=1}^{n}\gamma(x_i;\theta)$ or the worst-case margin: $\min_i \gamma(x_i;\theta)$. Let us recall the intuition illustrated in Figure 3. We would like the most points to be away from the decision bounday and such intuition suggest that the worst-case margin is a better choice. Mathematically, the hard-SVM solves the following maxmin problem:

$$\max_{\theta} \min_{i\in[n]} \gamma(x_i;\theta) \qquad \text{(Hard-SVM)} \tag{7}$$

By ensuring each sample to be classified correctly, i.e., $\beta^T x_i + \beta_0$ and y_i should have the same sign. Then, the problem (Hard-SVM) is equivalent to

$$\max_{\beta\in\mathbb{S}^{d-1},\beta_0\in\mathbb{R}} \min_{i\in[n]} y_i(\beta^T x + \beta_0). \tag{8}$$

The above is a minimax problem, which is not easy to optimize. It can be converted into a constraint optimization problem as follows.

Theorem 3.1. *The problem* (8) *is eqivalent to*

$$\begin{aligned} \min_{\beta,\beta_0} \ & \|\beta\|^2 \\ s.t. \ & y_i(\beta^T x_i + \beta_0) \geq 1, \forall i \in [n] \end{aligned} \tag{9}$$

Proof. Let $r = \min_i y_i(\beta^T x_i + \beta_0)$. Then (8) can be solved by

$$\max_{\beta\in\mathbb{S}^{d-1},\beta_0\in\mathbb{R}} r$$

$$s.t. \quad y_i(\beta^T x_i + \beta_0) \geq r, \ \forall i \in [n].$$

Let $\tilde{\beta} = \beta/r, \tilde{\beta}_0 = \beta_0/r$. Then, the problem becomes

$$\begin{aligned} &\min_{\tilde{\beta}} \|\tilde{\beta}\| \\ &\quad s.t. \ y_i(\tilde{\beta}^T x_i + \tilde{\beta}_0) \geq 1, \ \forall i \in [n]. \end{aligned}$$

□

Remark 3.2. The above procedure is a very useful trick in convex optimization.

The problem (9) can be solved efficiently by using convex optimization methods. Essentially, (9) attempts to look for a minimum-ℓ_2-norm solution among all the classifiers that separate data points with all sample margins no less than 1.

General max-margin classifier Given a function class $\mathcal{F}$ that satisfies $\alpha f \in \mathcal{F}$ for any $f \in \mathcal{F}, \alpha > 0$, we say the data are $\mathcal{F}$-separable if there exist $f \in \mathcal{F}$ such that

$$\min_i y_i f(x_i) \geq 1.$$

Definition 3.3. Given a classifier f, we define $\gamma_f(x) = yf(x)$ as the margin of f at x.

In such a general case, one can intuitively interpret "margin" as the confidence of prediction correctness. In the linear case, if choosing $\Omega(\cdot)$ to be the ℓ_2 norm of parameters, then $\gamma_f(x)$ is equal to the Euclidean distance from x to the decision boundary. However, in general, the margin might not have a clear geometric interpretation.

The max-margin classifier can be defined as the solution of the following problem

$$\begin{aligned} &\min_{f \in \mathcal{F}} \ \Omega(f) \\ &\quad s.t. \ y_i f(x_i) \geq 1, \ i \in [n], \end{aligned} \tag{10}$$

where $\Omega(\cdot)$ is a penalization used to control the complexity of decision function. This recovers the hard-SVM if f is linear and $\Omega(\cdot)$ is the ℓ_2 norm.

3.2 Soft-SVM

In the above derivation, we assume that the data are well-separated. The soft-SVM extends this to the non-separable case. Introduce the slack variable $\xi_i \geq 0$ to denote the margin violation at the i-th sample. Then, the soft-SVM solves the following optimization problem:

$$\begin{aligned} \min_{f \in \mathcal{F}, \xi \in \mathbb{R}^n} \quad & \lambda \Omega(f) + \frac{1}{n} \sum_{i=1}^n \xi_i \\ s.t. \quad & y_i f(x_i) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \qquad \forall i \in [n]. \end{aligned} \tag{11}$$

This means we allow the margin of each sample to have a violation ξ_i, but we minimize the average violation when selecting classifier.

Lemma 3.4. *The problem* (11) *is equivalent to*

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

where $\ell(z) = \max(0, 1-z)$ *is the hinge loss.*

Proof. Notice that (11) is linear in $(\xi_1, \ldots, \xi_n)$. Specifically, fixing f and minimizng the objective with respect to $\xi_1, \ldots, \xi_n$ gives the smallest violations for any fixed $f \in \mathcal{F}$:

$$\xi_i(f) = \begin{cases} 0 & \text{if } y_i f(x_i) \geq 1 \\ 1 - y_i f(x_i) & \text{if } y_i f(x_i) < 1. \end{cases}$$

Obviously, $\xi_i(f) = \max(0, 1 - y_i f(x_i))$. Plugging it back, we have the problem becomes

$$\min_{\theta} \frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - y_i f(x_i)) + \lambda\Omega(f).$$

□

We have the following observations for the hinge loss:

- If the margin is less than 1, a linear penalization is imposed.
- If the margin is greater than 1, no penalization is imposed.

The second point is very important, which allows soft-SVM to recover the hard-SVM when data are linearly separated (need also to take $\lambda \to 0$). In addition, it also makes much sense intuitively since one should not impose any or too much penalization if the predictions are correct and confident.

3.3 A margin-based approach of designing surrogate loss

The fundamental insights behind the soft-SVM are: we should only penalize when the margin (i.e., the prediction confidence) is relatively small. The hinge loss is only one choice and we can also design other surrogate losses to achieve similar effects.

- Squared hinge loss: $\ell(z) = \max(0, 1-z)^2$. It imposes quadratic penalization for small and negative margins, and no penalization if margins are significantly positive.
- Exponential loss: $\ell(z) = e^{-z}$. Exponentially-small penalization if the margin is significantly positive; exponentially-large penalization for negative margins.
- Logistic loss: $\ell(z) = \log(1+e^{-z})$. This corresponds to the softmax classifier for binary classification. It has the following tails:

  $$\ell_{logistic}(z) \approx \begin{cases} e^{-z} & \text{if } z \gg 1 \\ -z & \text{if } z \ll -1, \end{cases}$$

  This means that logistic loss behaves like a mixture of hinge loss and exponential loss: exponentially small penalization if the margin is significantly large; linear penalization if the margin is relatively negative.

Figure 4 provides a visual comparision among different loss functions.

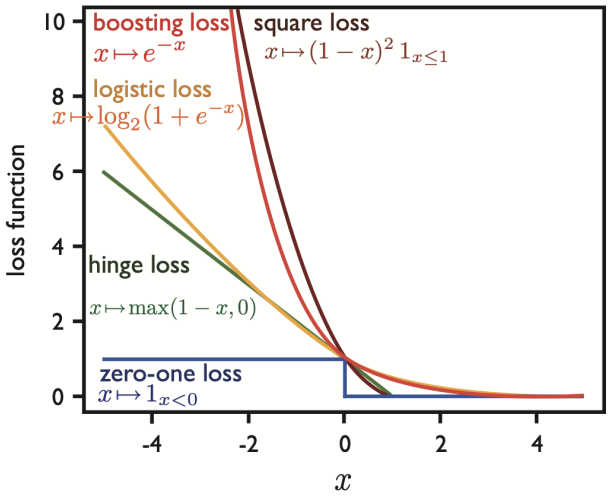

Figure 4: Example of loss functions

4 Representer theorem and the feature space SVM

The idea behind hard-SVM highly relies on linear separation. Another way of extending hard-SVM is to lift the input to a (higher-dimensional) feature space; then these data could be more linearly separable in the feature space. We have seen similar ideas in kernel methods. In fact, the popularity of kernel methods is driven by the development of SVM classifiers. Figure 5 provides an example, showing that data are linearly separated in feature space but not in the original space.

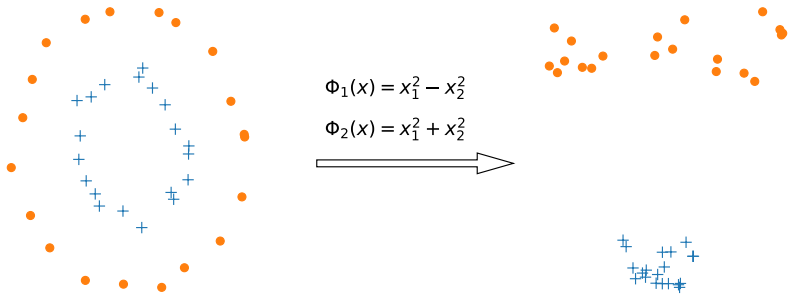

Figure 5: An illustration of how nonlinear feature maps can make the data linearly separated in the feature space. Then we can apply hard-SVM to the feature space.

Let $\Phi : \mathbb{R}^d \mapsto \mathcal{H}$ be a feature map and $\mathcal{H}$ be the feature space. The associated kernel $k(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{H}}$. The model is $f_\beta(x) = \langle \beta, \Phi(x) \rangle_{\mathcal{H}}$. Consider the soft-SVM in the feature space:

$$\min_{\beta} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i f_\beta(x_i)) + \lambda \|\beta\|_{\mathcal{H}}^2 \tag{12}$$

Let $V = \text{span}\{\Phi(x_1), \ldots, \Phi(x_n)\} \subset \mathcal{H}$ and consider the orthogonal decomposition

$$\beta = \beta_{\|} + \beta_{\perp},$$

where $\beta_{\|} \in V$ and $\beta_{\perp} \in V^{\perp}$. Since $\|\beta\|^2 \geq \|\beta^{\|}\|^2$ and $f_{\beta}(x_i) = f_{\beta_{\|}}(x_i)$ for any $i \in [n]$, we have

$$\min_{\beta} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i f_{\beta}(x_i)) + \lambda \|\beta\|_{\mathcal{H}}^2 = \min_{\beta \in V} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i f_{\beta}(x_i)) + \lambda \|\beta\|_{\mathcal{H}}^2$$

Thus, the infinite-dimensional optimization problem is reduced to a finite-dimensional problem.

Let $\beta = \sum_{j=1}^{n} \alpha_j \Phi(x_j)$. Then, the objective becomes

$$\begin{aligned} &\min_{\alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} \ell(\sum_{j=1}^{n} \alpha_j y_i \langle \Phi(x_i), \Phi(x_j) \rangle) + \lambda \sum_{i,j} \alpha_i \alpha_j \langle \Phi(x_i), \Phi(x_j) \rangle_{\mathcal{H}} \\ \Longrightarrow &\min_{\alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i e_i^T K \alpha) + \lambda \alpha^T K \alpha, \end{aligned} \tag{13}$$

where $K = (k(x_i, x_j)) \in \mathbb{R}^{n \times n}$ is the kernel matrix. This problem only needs to access the kernel.

Note that the above derivation works for general loss functions. In particular, the kernel-version SVM corresponds to the case of choosing the hinge loss. In fact, the above derivation also applies to the original space, where the kernel is given by $k(x, x') = x^T x'$.

4.1 What are support vectors?

Let $\alpha^* \in \mathbb{R}^n$ be the solution of (13). Then, the decision boundary is determined by

$$\beta^* = \sum_{j} \alpha_j^* \Phi(x_j). \tag{14}$$

The support vectors correspond to x_j's that are activated, i.e., the set $\{x_j : \alpha_j^* \neq 0\}$. The decision boundary is completely determined by these support vectors. In the theory of SVM, an important quantity , related to the generalization performance, is the number of support vectors: $n_s = |\{x_j : \alpha_j^* \neq 0\}|$.

Let us consider the linearly-separable case illustrated in the left panel of Figure 6. In such a case, the use of hinge loss "recovers" the hard-SVM when $\lambda \to 0$. Thus the support vectors only correspond to *the points that are closest to the decision boundary* (in other words, the points with smallest margins) [1]. In such a case, we have that $n_s \ll n$ and this fewness of support vectors is also important to guarantee the good generalization of a classifier as explained in Section 3.1.

However, in moderm applications, it is also very often that the data may distribute like the right panel of Figure 6, where nearly all points are close to the decision boundary. In such a case, looking for a decison boundary with fewest support vectors might be not necessary. But it is always preferred to look for a decision boundary such that all data points are away from it as much as possible. This may partially explain why the hinge loss is not very popular in solving complex problems such as image recognition.

5 Softmax classifier is approximately max-margin

We first need some properties of the softmax operator.

[1] The rigorous justification of this statement is left to homework.

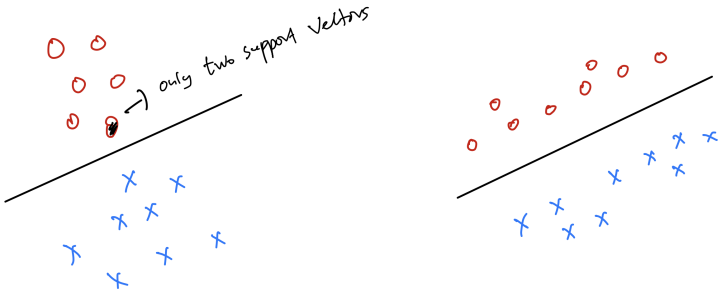

Figure 6: **Left:** A typical situation where the softmax classifier is roughtly max-margin. In such a case, the number of support vectors are small, i.e., $n_s \ll n$, and thus, the max-margin classifier generalize well. **Right:** A situation where there is not clear margin-gap separation and the number of support vectors are close to the number of samples. Note that in this case, nearly all data points are close to the decision boudary.

Lemma 5.1. *Let $k = \operatorname{argmax}_j z_j$. Then, as $T \to 0$, we have*

$$\left(\frac{e^{z_1/T}}{\sum_j e^{z_j/T}}, \frac{e^{z_2/T}}{\sum_j e^{z_j/T}}, \ldots, \frac{e^{z_C/T}}{\sum_j e^{z_j/T}} \right) \to (0, \ldots, 0, 1, 0 \ldots, 0) = e_k.$$

In other words, as the temperature $T \to 0$, the softmax becomes the max operator.

Proof. Omited! □

Remark 5.2. Note that the max operator is not continuous and thus replaced with a smoothed max operator is very useful in many applications. The softmax is the most popular one but there exist other choices; see https://en.wikipedia.org/wiki/Smooth_maximum.

The following lemma provides a non-asymptotic bound.

Lemma 5.3 (The LogSumExp trick)**.** *Let $z_1, \ldots, z_n \in \mathbb{R}$. For any $T > 0$, we have*

$$\max_j z_j \le T \log \sum_{j=1}^{n} e^{z_j/T} \le \max_j z_j + T \log(n).$$

The proof is left to homework.

Theorem 5.4. *Let $\ell(z) = e^{-z/T}$ and $\Omega(\cdot)$ be a complexity measure. Let $\hat{\theta}_T = \min_{\Omega(\theta)\le 1} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i f_\theta(x_i))$. Let $\gamma(\theta) = \min_{i\in[n]} y_i f_\theta(x_i)$ be the worst-case margin and*

$$\bar{\theta} = \max_{\Omega(\theta)\le 1} \gamma(\theta) \tag{15}$$

be the max-margin solution. Then,

$$\gamma(\bar{\theta}) - T \log n \le \gamma(\hat{\theta}_T) \le \gamma(\bar{\theta}).$$

This theorem shows that the margin of softmax classifier is close to the one of max-margin classifier, in particular when $T \ll 1$.

Proof. Let $Q(\theta) = -T \log \left(\sum_{i=1}^n e^{-y_i f_\theta(x_i)/T} \right)$ be the softmax margin. Then,

$$\hat{\theta}_T = \operatorname*{argmin}_\theta T \log \left(\sum_{i=1}^n e^{-y_i f_\theta(x_i)/T} \right) = \operatorname*{argmin}_\theta (-Q(\theta)).$$

By Lemma 5.3, we have

$$\max_i (-y_i f_\theta(x_i)) \le -Q(\theta) \le \max_i (-y_i f_\theta(x_i)) + T \log n,$$

leading to

$$\gamma(\theta) - T \log n \le Q(\theta) \le \gamma(\theta).$$

By the deinition of $\bar{\theta}$ and $\hat{\theta}_T$, we have $\gamma(\hat{\theta}_T) \le \gamma(\bar{\theta})$ and

$$\gamma(\bar{\theta}) - T \log n \le Q(\bar{\theta}) \le Q(\hat{\theta}_T) \le \gamma(\theta_T).$$

Thus, we complete the proof. □

In the above theorem, we only consider the exponential loss for simplicity. In fact, the same observation holds for any surrogate loss with an exponential tail if taking T to be sufficiently small. The cross-entropy loss (i.e., the logisitic loss for binary classification) is a particular example.

We also remark that the closeness implied by the theorem above needs T to be sufficiently small. In fact, for many situations, the data implicitly introduce a "intrinsic" temperature. Let $\gamma_i(\theta) = y_i f_\theta(x_i)$ and $k = \operatorname{argmax}_i \gamma_i(\theta)$. Then, if $\gamma_k(\theta) \gg \gamma_i(\theta)$ for $i \neq k$ (this corresponds to the left of Figure 6), we have

$$\max_\theta \sum_i e^{-\gamma_i(\theta)} = \max_\theta e^{-r_k(\theta)} (1 + \sum_{i \neq k} e^{-(\gamma_k(\theta) - \gamma_i(\theta))}) \approx \max_\theta e^{-\gamma_k(\theta)},$$

where the approximation is exponentially accurate with respect to the margin gaps $\{\gamma_k(\theta) - \gamma_i(\theta)\}_{i \neq k}$. This implies that minimization of the exponential loss roughy gives rise to the max-margin classifier, as long as long the margin gaps are sufficiently large.

For the right plot of Figure 6 where the margin gaps are small, the softmax classifier does not behave like a max-margin classifier unless we take the temperature T to be sufficiently small.