Privacy in Statistics and Machine Learning — Spring 2021
Lecture 11: Empirical Risk Minimization

Adam Smith and Jonathan Ullman

1 Optimization for Fitting Models

For many natural problems in machine learning and statistics, the output we desire can phrased as minimizing some loss function defined by the data set.

For example, the mean of a set $\mathbf{x}$ of numbers $x_1, ..., x_n \in R$ is the number μ that minimizes the sum of the squares[1] of the differences between μ and the x_i's:

$$\mu(\mathbf{x}) = \arg\min_{w\in\mathbb{R}} L(w;\mathbf{x}) \quad \text{where} \quad L(w;\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}(w-x_i)^2 .$$

The notation "$\arg\min_{w\in\mathcal{W}} L(w)$" denotes a minimizer of the function L in the set $\mathcal{W}$, if a minimizer exists. If there is no minimizer, such as in the expression "$\arg\min_{w\in\mathbb{R}} w$" or "$\arg\min_{w\in(0,1)} \ln(w)$", then the notation is not defined.

Similarly, one way to define the median is as a minimizer of the function $L(w;\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}|w-x_i|$. This view of the median comes up in the exponential-mechanism algorithm for the median developed in Homework 2.

In classical problem of ordinary least squares linear regression, each data point is a pair (x_i, y_i) where $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$. Our goal is to find a a vector w in $\mathbb{R}^d$ such that $\langle w, x_i\rangle \approx y_i$ for all i, where $\langle w, x\rangle$ denotes the inner product $\langle w, d\rangle = \sum_{j=1}^{d} w(j)\cdot x(j)$. Specifically, we seek to minimize

$$L(w;\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\left(\langle w, x_i\rangle - y_i\right)^2 .$$

Empirical Risk Minimization (ERM) for Decomposable Losses These examples fit a general framework: there is a loss function $L(w;\mathbf{x})$ which takes a parameter vector w and a dataset $\mathbf{x} \in \mathcal{U}^n$. Many loss functions arising in statistics and ML are *decomposable*, meaning they can be written as a sum of terms, where each term depends on at most one of the x_i's, as follows:

$$L(w;\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\underbrace{\ell(w;x_i)}_{\text{"individual losses"}} + \underbrace{\Lambda(w)}_{\text{"regularizer"}} . \tag{1}$$

The terms in the sum are called individual losses and the last term $R(w)$, which depends only on w but not the data, is called the *regularizer*.

In general, suppose we are given a loss function ℓ and a feasible set $C \subseteq \mathbb{R}^d$ of acceptable parameters w (for example, we might need the entries of w to be nonegative, or we might insist on a solution with

[1] To see why the mean really minimizes sum of squared distances, notice that the derivative of L with respect to w is $L'(w) = \sum_{i=1}^{n} 2(w-x_i) = 2(nw - \sum_{i=1}^{n} x_i)$. This last expression is 0 exactly when w is the mean. Since L is differentiable everywhere and increases when w tends to either $-\infty$ or $+\infty$, the value where the derivative is 0 is the unique minimizer of L.

norm $\|w\| \leq 1$. The problem of minimizing $L(w; \mathbf{x})$ on C is called *empirical risk minimization.* Given an output $\hat{w}$, we measure success by the "excess risk",

$$\text{Excess empirical risk at } \hat{w}: \quad L(\hat{w}) - \min_{w \in C} L(w; \mathbf{x}) \tag{2}$$

Population Risk We will focus hereon empirical risk, but it often makes sense to also analyze the population risk, or "generalization" of a solution. Suppose the data as drawn from some unknown population, modeled by a distribution P. In that case, we may also consider how well our solution $\hat{w}$ does with respect to unseen data drawn from the same distribution.

$$\text{Population loss:} \quad L(w; P) \overset{\text{def}}{=} \mathop{\mathbb{E}}_{x \sim P} (L(w; \mathbf{x})). \tag{3}$$

$$\text{Excess population risk at } \hat{w}: \quad L(\hat{w}; P) - \min_{w \in C} L(\hat{w}; P) \tag{4}$$

We will return to population risk and generalization later in the course.

Further Examples In a support vector machine (SVM), the data are pairs $(x_i, y_i) \in \mathbb{R}^d \times \{-1, 1\}$ and we aim to minimize

$$L(w; \mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} (1 - y_i \langle w, x_i \rangle)_+ + \lambda \|w\|_2^2, \tag{5}$$

where $(a)_+$ is shorthand for $\max(0, a)$, and λ is a parameter that helps to select for "simple" (short) solutions. The larger λ is, the more we penalize long solutions.

The individual loss here is called the *hinge loss.* The idea is that w defines a classifier given by $\text{sign}(\langle w, x_i \rangle)$. The loss function here, called the hinge loss, applies no penalty at all when the sign of $\langle w, x_i \rangle$ is correct (that is, equal to y_i) and the absolute value of $\langle w, x_i \rangle$ is at least 1. Otherwise it applies a penalty that changes gradually with $\langle w, x_i \rangle$. This loss is intended to be a continuous, convex alternative to the misclassification losss, which would simply be 1 if $y_i \langle w, x_i \rangle > 0$ and 0 otherwise.

Neural nets provide another rich class of decomposable losses. The entries of w represent weights in the network, and one of several loss functions is used to penalize weights that lead to predictions that do not fit the data well.

The examples we've given are for continuous loss functions (i.e. where ℓ is continuous in w), and these will continue to be our focus. The general framework makes sense for discrete problems, but the tools one employs and the flavor of the algorithms are different.

Losses and Likelihoods Many loss functions used in practice are derived from some probabilistic model of data generation. Specifically, we often think of the parameters w as defining a distribution on the whole data (so $p(x|w)$ is a valid distribution on x for each w) or on the label (so that $p(y|x, w)$ is a valid distribution on y for every fixed w and x.

Given a data set that an analyst conjectures has been generated according to a given model, a natural approach is to find the parameter w that would have had the highest probability of generating $\mathbf{x} = (x_1, ..., x_n)$, assuming the data were generated i.i.d.. We can write that probability as a product $\prod_{i=1}^{n} p(x_i|w)$. Taking logarithms, we can turn the maximization problem into a sum, which makes calculations easier:

$$w \in \arg\max \prod_{i=1}^{n} p(x_i|w) \qquad \Longleftrightarrow \qquad w \in \arg\min L(w; \mathbf{x}) = \sum_{i=1}^{n} \ell(w; x_i) \text{ where } \ell(w; x) = \log \frac{1}{p(x|w)}.$$

This value of w is called the *maximum likelihood estimator* for the probability model.

For example, if we think that the data were drawn from a Gaussian distribution with variance 1 and unknown mean μ, the mean of the data set is the maximum likelihood estimator for μ. The median corresponds to maximum likelihood estimation for the Laplace distribution (with known variance and unknown mean).

2 Private ERM

Given a loss function ℓ specifying a decomposable loss L and a feasible set C, we can ask for a differentially private algorithm that solves the ERM problem. Without the privacy constraint, we could hope for a solution with zero excess empirical risk. The randomness inherent to differential privacy makes that impossible in general, but we could potentially bound the excess risk, either in expectation or with high probability. For simplicity, we'll focus on excess empirical risk.

In order to solve the problem differentially privately, we'll need to somehow bound the influence of any one data point on the loss function L. We'll consider two assumptions on the individual loss function $\ell : C \times \mathcal{U} \to \mathbb{R}$:

- *Bounded loss:* We say the loss ℓ is Δ-bounded if for every data value $x \in \mathcal{U}$ and for every $w \in C$,

  $$\ell(w;x) \in [0, \Delta] .$$

  This is essentially the same as asking that the overall loss function $L(w;\cdot)$ have global sensitivity at most Δ for every fixed w.

  For example, in classification problems, the *misclassification loss* is always bounded.
- *Lipschitz loss:* The individual loss ℓ is G-Lipschitz if for every data value $x \in \mathcal{U}$ and for every $w \in C$, the gradient of ℓ with respect to w is bounded by G:

  $$\|\nabla \ell(w;x)\|_2 \le G .$$

  In fact, the loss doesn't need to be differentiable to be Lipschitz—the more general definition is that for every x and every two vectors v, w, we have $|\ell(v;x) - \ell(w;x)| \le G\|v - w\|$. This is implied by an upper bound on the gradient (why?) but allows for a wider range of functions. For example, the loss function defining the median $\ell(w;x) = |w - x|$ is 1-Lipschitz. The hinge loss (5) is G-Lipschitz as along as we restrict the length of the data vectors x to be at most G, since by the chain rule[2] $\nabla \ell(w;x) = \left(\frac{d}{dt}(1-t)_+ \Big|_{t=\langle w,x \rangle} \right) \cdot x$. The first term is -1 or 0, so the gradient's norm is at most $\|x\|_2$.

Exercise 2.1. Show that if the feasible set C is bounded, then Lipschitz loss functions are bounded. Specifically, if ℓ is G-Lipschitz and the diameter of C is R, then ℓ is Δ-bounded for $\Delta = G \cdot R$.

Exercise 2.2. Under what conditions on C and $\mathcal{U}$ is the linear regression loss bounded? Lipshitz?

Exercise 2.3. Show that if the individual loss ℓ is G-Lipschitz, then so is the overall loss L.

[2] Recall the chain rule from calculus: if $f, g : \mathbb{R} \to \mathbb{R}$ are differentiable, then $\frac{d}{dw} f(g(w)) = \left(\frac{d}{dt} f(t) \Big|_{t=g(w)} \right) \cdot \frac{d}{dw} g(w)$.

2.1 A First Algorithm via the Exponential Mechanism

Suppose we have a bounded individual loss function $\ell : C \times \mathcal{U} \to [0, \Delta]$ together with an arbitrary regularizer. One way to solve the ERM problem is via the exponential mechanism:

Algorithm 1: ExpMech-ERM($\ell(\cdot;\cdot), \varepsilon, \mathbf{x}$)

```
Input: Assume $\ell : C \times \mathcal{U} \to [0, \Delta]$ and $\mathbf{x} \in \mathcal{U}^n$
Define $L(w; \mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} \ell(w; x_i)$ ;
Select $\hat{W}$ from the distribution on $C$ with $\Pr(W = w) \propto \exp\left(-\frac{\varepsilon n}{2\Delta} L(w; \mathbf{x})\right)$;
return $\hat{W}$;
```

This algorithm is well-defined when C and ℓ are not too pathological.[3] We'll assume going forward that the algorithm makes sense.

Lemma 2.4. *ExpMech-ERM (Algorithm 1) is $(\varepsilon, 0)$-DP.*

The lemma follows directly from the fact that ℓ is Δ-bounded, so the overall loss function L is $\frac{\Delta}{n}$-sensitive.

One can prove a number of different types of utility theorems about this algorithm. We go with a relatively straightforward one:

Theorem 2.5. *Suppose C is the ℓ_2-ball of radius R in $\mathbb{R}^d$ and ℓ is G-Lipschitz. Then for every data set $\mathbf{x}$, ExpMech-ERM with $\Delta = GR$ returns a parameter $\hat{W}$ such that $\mathbb{E}\left(L(\hat{W}; \mathbf{x})\right) \leq \min_{w \in C} L(w; \mathbf{x}) + O\left(\frac{dGR \log(\frac{\varepsilon n}{d})}{\varepsilon n}\right)$.*

For constant G and R, this gives us small excess risk roughly when $n = \omega(d/\varepsilon)$. That seems like a good start—when we have more data than dimensions, we can get a good answer!

The main drawback of this algorithm is that it requires sampling from a pretty odd distribution. Even when the set C is a ball, there is no reason to think that we can sample in polynomial time. For example, the loss functions defined by deep neural nets would lead to distributions for which there are no good algorithms that would be guaranteed to converge.

In fact, we can run this algorithm in polynomial time when the loss function we are optimizing is *convex*. It turns out we can also tighten the analysis when L is convex to get rid of the logarithmic factor. Unfortunately, the sampling algorithms for general convex functions are not really practical. In the next lecture, we'll talk more about convex functions and see a family of algorithms based on gradient descent that are both more efficient and more accurate. But first let's prove the theorem.

Proof. Fix a dataset $\mathbf{x}$. We will use $L(w)$ as shorthand for $L(w; \mathbf{x})$. Let w^* be a true minimizer of $L(w)$ in C. Fix a small radius r, which we will set later to be about $d/(\varepsilon n)$. To bound the probability of getting an output with high empirical risk, consider two sets of outputs:

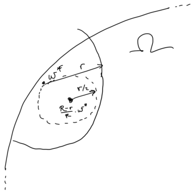
$$\begin{aligned} GOOD &= \{w \in C \,:\, L(w) \leq L(w^*) + rG\} \\ BAD &= \{w \in C \,:\, L(w) \geq L(w^*) + rG + t\} \end{aligned}$$

Now $\mathbb{P}_{\hat{W}}\,(BAD)$ is at most

$$\frac{\mathbb{P}_{\hat{W}}\,(BAD)}{\mathbb{P}_{\hat{W}}\,(GOOD)} \leq \frac{\mathrm{Vol}(BAD)\exp(\frac{-\varepsilon n}{2RG}(L(w^*) + rG + t))}{\mathrm{Vol}(GOOD)\exp(\frac{-\varepsilon n}{2RG}(L(w^*) + rG))} = \frac{\mathrm{Vol}(BAD)}{\mathrm{Vol}(GOOD)} e^{-\frac{\varepsilon n}{2RG} t} .$$

[3] Say, when C is bounded and contains a non-empty ball and R and ℓ are measurable.

The BAD set's volume is at most that of the entire ball of radius R. But what about GOOD? Since ℓ is G-Lipschitz, so is L. Thus $GOOD$ contains all the points within distance r of w^* that lie within C. Even if w^* is right up against the boundary of C, the ball of radius r around w^* contains a ball of radius $r/2$ that is entirely within C (namely, the ball centered at $\frac{R-r}{R}w^*$).



So the ratio is $\frac{\mathrm{Vol}(BAD)}{\mathrm{Vol}(GOOD)}$ is at most the ratio of the volumes of the balls of radius R and $r/2$. That ratio is $\left(\frac{2R}{r}\right)^d$. The rest of the proof is simply putting these pieces together. We now know

$$\mathbb{P}(BAD) \leq \left(\frac{2R}{r}\right)^d \cdot \exp\left(-\frac{\varepsilon n}{2RG}t\right) = \exp\left(d \ln\left(\frac{2R}{r}\right) - \frac{\varepsilon n}{2RG}t\right)$$

For any $\beta > 0$, we can set $t = \frac{2RG}{\varepsilon n}\big(d\ln(\frac{2R}{r}) + \ln(1/\beta)\big)$, to get that the probability of BAD is at most β.

Now the elements of BAD have excess risk $rG + t$. Setting $r = 2R \cdot \frac{d}{n\varepsilon}$, and t as above, we get that with probability at most β, the excess risk of $\hat{W}$ is at most

$$2RG\left(\frac{d}{\varepsilon n} + \frac{d\ln(\varepsilon n/d) + \ln(1/\beta)}{\varepsilon n}\right).$$

Since this holds for every $\beta > 0$, we can use the integral formula for expectation[4] to wrap up the proof. □

The theorem can actually be proven for any convex set C of diameter R, not just the Euclidean ball. We'll discuss convex sets and functions in the next lecture.

3 Convexity

Before the next lecture, it would help to brush up on some basics on convex sets and functions.

Definition 3.1. A set C in $\mathbb{R}^d$ is convex if every two points $x, y \in C$ can "see each other", that is, if the line segment from x to y is entirely within C.

For example, cubes, pyramids, and spheres are convex, but "donuts" (tori) and other shapes with holes in them are not.

Definition 3.2. A function $f : C \to \mathbb{R}$ defined on a convex set $C \subseteq \mathbb{R}$ is convex if for every two points $x, y \in C$, we have

$$f\left(\frac{x+y}{2}\right) \leq \frac{f(x) + f(y)}{2}.$$

For example, $f(x) = x^2$ and $f(x) = |x|$ and $f(x) = \ln(1/x)$ (where it is defined) are convex, but $f(x) = (x-1)^3$ is not (why?).

This definition is clean and simple, but not actually that easy to work with. An equivalent, and much more useful, definition of convexity is the following:

[4] $\mathbb{E}(Z) = \int_{z \geq 0} \Pr(Z \geq z) dz$ for nonnegative random variables Z.

Lemma 3.3. *Given a convex set C, a function $f : C \to \mathbb{R}$ is convex if and only if: for every point $x \in C$, we can find an* affine *function g_x such that $f(x) = g_x(x)$ and $f(y) \geq g_x(y)$ for all $y \in C$.*

When f is differentiable at x, the function g_x is just the first-order Taylor approximation

$$g_x(y) = f(x) + \langle \nabla f(x), y - x \rangle .$$

However, the lemma makes sense even when f is not differentiable at x. In that case, we get many possible functions g_x that are valid lower bounds for f. For instance, at $x = 0$, the absolute value function $f(x) = |x|$ can be bounded below by $g_x(y) = cy$ for any constant c between -1 and 1. In general, the set of affine functions that are valid lower bounds for f at x define the subgradient set of f at x

$$\partial f(x) \overset{\text{def}}{=} \{w \ : \ (\forall y \in C) f(x) + \langle w, y - x \rangle \leq f(y)\} . \tag{6}$$

Exercise 3.4. Use Lemma 3.3 to prove *Jensen's inequality*: if f is a convex function defined on a convex set C, then for every random variable X taking values in C,

$$\mathbb{E}\,(f(X)) \geq f(\mathbb{E}\,(X)) .$$