[Lecture 8]

0.1 Efficient Algorithms

We now turn our attention to efficient algorithms. Recall that previously we considered minimum distance functionals projecting onto sets $\mathcal{G}$ and $\mathcal{M}$ under distances TV and $\widehat{\mathsf{TV}}$. Here we will show how to approximately project onto the set $\mathcal{G}_{\mathsf{cov}}(\sigma)$, the family of bounded covariance distributions, under TV distance. The basic idea is that if the true distribution p^* has bounded covariance, and $\tilde{p}$ does not, the largest eigenvector of $\mathsf{Cov}_{\tilde{p}}[X]$ must be well-aligned with the mean of the bad points, and thus we can use this to remove the bad points. If on the other hand $\tilde{p}$ has bounded covariance, then its mean must be close to p^* by our previous modulus bounds and so we are already done.

To study efficient computation we need a way of representing the distributions $\tilde{p}$ and p^*. To do this we will suppose that $\tilde{p}$ is the empirical distribution over n points $x_1, \ldots, x_n$, while p^* is the empirical distribution over some subset S of these points with $|S| \geq (1-\epsilon)n$. Thus in particular p^* is an ϵ-deletion of $\tilde{p}$.

Before we assumed that $\mathsf{TV}(p^*, \tilde{p}) \leq \epsilon$, but taking $p' = \frac{\min(p^*, \tilde{p})}{1-\mathsf{TV}(p^*,\tilde{p})}$, we have $p' \leq \frac{\tilde{p}}{1-\epsilon}$ and $\|\mathsf{Cov}_{p'}[X]\| \leq \frac{\sigma^2}{1-\epsilon} \leq 2\sigma^2$ whenever $\|\mathsf{Cov}_{p^*}[X]\| \leq \sigma^2$. Therefore, taking $p^* \leq \frac{\tilde{p}}{1-\epsilon}$ is equivalent to the TV corruption model from before for our present purposes.

We will construct an efficient algorithm that, given $\tilde{p}$, outputs a distribution q such that $\mathsf{TV}(q, p^*) \leq \mathcal{O}(\epsilon)$ and $\|\mathsf{Cov}_q[X]\|_2 \leq \mathcal{O}(\sigma^2)$. This is similar to the minimum distance functional, in that it finds a distribution close to p^* with bounded covariance; the main difference is that q need not be the projection of $\tilde{p}$ onto $\mathcal{G}_{\mathsf{cov}}$, and also the covariance of q is bounded by $\mathcal{O}(\sigma^2)$ instead of σ^2. However, the modulus of continuity bound from before says that *any* distribution q that is near p^* and has bounded covariance will approximate the mean of p^*. Specifically, we have

$$\|\mu(q) - \mu(p^*)\|_2^2 \leq \mathcal{O}(\max(\|\mathsf{Cov}_q[X]\|, \|\mathsf{Cov}_{p^*}[X]\|) \cdot \mathsf{TV}(p^*, q)) = \mathcal{O}(\sigma^2\epsilon). \tag{1}$$

We will show the following:

Proposition 0.1. *Suppose $\tilde{p}$ and p^* are empirical distributions as above with $p^* \leq \tilde{p}/(1-\epsilon)$, and further suppose that $\|\mathsf{Cov}_{p^*}[X]\| \leq \sigma^2$. Then given $\tilde{p}$ (but not p^*), there is an algorithm with runtime $\mathrm{poly}(n, d)$ that outputs a q with $\mathsf{TV}(p^*, q) \leq \epsilon$ and $\|\mathsf{Cov}_q[X]\| \leq O(\sigma^2)$. In particular, $\|\mu(p^*) - \mu(q)\|_2 = \mathcal{O}(\sigma\sqrt{\epsilon})$.*

Note that the conclusion $\|\mu(p^*) - \mu(q)\|_2 \leq \mathcal{O}(\sigma\sqrt{\epsilon})$ follows from the modulus bound on $\mathcal{G}_{\mathsf{cov}}(\sigma)$ together with the property $\mathsf{TV}(p^*, q) \leq \epsilon$.

The algorithm, `FilterL2`, underlying Proposition 0.1 is given below; it maintains a weighted distribution $q(c)$, which places weight $c_i / \sum_{j=1}^n c_j$ on point x_i. It then computes the weighted mean and covariance, projects onto the top eigenvector, and downweights points with large projection.

Algorithm 1 `FilterL2`

1: Input: $x_1, \ldots, x_n \in \mathbb{R}^d$.
2: Initialize weights $c_1, \ldots, c_n = 1$.
3: Compute the empirical mean $\hat{\mu}_c$ of the data, $\hat{\mu}_c \stackrel{\text{def}}{=} (\sum_{i=1}^n c_i x_i)/(\sum_{i=1}^n c_i)$.
4: Compute the empirical covariance $\hat{\Sigma}_c \stackrel{\text{def}}{=} \sum_{i=1}^n c_i (x_i - \hat{\mu}_c)(x_i - \hat{\mu}_c)^\top / \sum_{i=1}^n c_i$.
5: Let v be the maximum eigenvector of $\hat{\Sigma}_c$, and let $\hat{\sigma}_c^2 = v^\top \hat{\Sigma}_c v$.
6: If $\hat{\sigma}_c^2 \leq 20\sigma^2$, output $q(c)$.
7: Otherwise, let $\tau_i = \langle x_i - \hat{\mu}_c, v\rangle^2$, and update $c_i \leftarrow c_i \cdot (1 - \tau_i/\tau_{\max})$, where $\tau_{\max} = \max_i \tau_i$.
8: Go back to line 3.

The factor $\tau_{\max}$ in the update $c_i \leftarrow c_i \cdot (1 - \tau_i/\tau_{\max})$ is so that the weights remain positive; the specific factor is unimportant and the main property required is that each point is downweighted proportionally to τ_i. Note also that Algorithm 1 must eventually terminate because one additional weight c_i is set to zero in every iteration of the algorithm.

The intuition behind Algorithm 1 is as follows: if the empirical variance $\hat{\sigma}_c^2$ is much larger than the variance σ^2 of the good data, then the bad points must on average be very far away from the empirical mean (i.e., τ_i must be large on average for the bad points).

More specifically, note that $\tau_i = \langle x_i - \hat{\mu}_c, v^* \rangle^2$. Let $\tilde{\tau}_i = \langle x_i - \mu, v^* \rangle^2$, and imagine for now that $\tau_i \approx \tilde{\tau}_i$. We know that the average of $\tilde{\tau}_i$ over the good points is at most σ^2, since $\tilde{\tau}_i$ is the variance along the projection v^* and $\|\mathsf{Cov}_{p^*}[X]\| \le \sigma^2$. Thus if the overall average of the τ_i is large (say $20\sigma^2$), it must be on account of the bad points. But since there are not that many bad points, their average must be *quite* large–on the order of σ^2/ϵ. Thus they should be easy to separate from the good points. This is depicted in Figure 1.

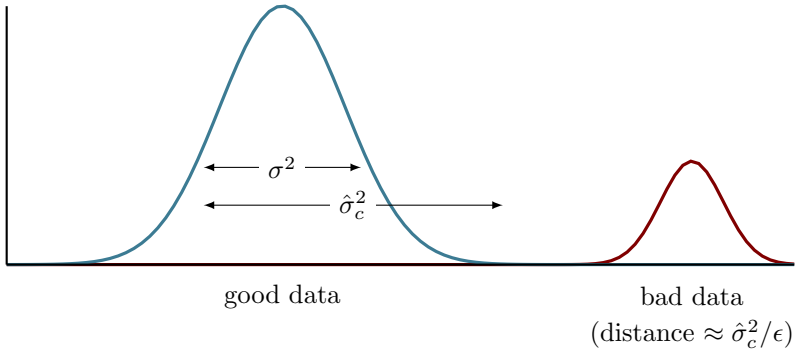

Figure 1: Intuition behind Algorithm 1. Because there is only an ϵ-fraction of bad data, it must lie far away to increase the variance by a constant factor.

This is the basic idea behind the proof, but there are a couple issues with this:

- The assumption that $\tilde{\tau}_i \approx \tau_i$ is basically an assumption that $\mu \approx \hat{\mu}_c$ (which is what we are trying to show in the first place!).
- The bad points are not deterministically larger than the good points; they are only separated in expected value.
- There are many fewer bad points than good points, so they are harder to find.

We will deal with the first issue by showing that μ is close enough to $\hat{\mu}_c$ for the algorithm to make progress. The second issue is why we need to do soft downweighting rather than picking a hard threshold and removing all points with τ_i above the threshold. We will resolve the third issue by showing that we always remove more mass c_i from the bad points than from the good points when we update c_i. Intuitively, while there are only ϵ times as many bad points as good points, this is balanced against the fact that the mean of the bad points is $1/\epsilon$ times as large as the mean of the good points.

We next put this intuition together into a formal proof.

Proof of Proposition 0.1. As above, for weights $c_i \in [0,1]$, let $q(c)$ be the distribution that assigns weight $c_i / \sum_j c_j$ to point x_i. Thus when $c_i = 1$ for all i, we have $q(c) = \tilde{p}$. Our hope is that as the algorithm progresses $q(c)$ approaches p^* or at least has small covariance. We will establish the following invariant:

$$\mathsf{TV}(q(c), p^*) \le \frac{\epsilon}{1-\epsilon} \text{ for all weight vectors } c \text{ used during the execution of Algorithm 1.} \qquad (\mathcal{I}_1)$$

We will do this by proving the following more complex invariant, which we will later show implies ($\mathcal{I}_1$):

$$\sum_{i \in S} (1 - c_i) \le \sum_{i \notin S} (1 - c_i) \qquad (\mathcal{I}_2)$$

The invariant ($\mathcal{I}_2$) says that the total probability mass removed from the good points is less than the total probability mass removed from the bad points. A key lemma relates ($\mathcal{I}_2$) to the τ_i:

Lemma 0.2. *If ($\mathcal{I}_2$) and $\sum_{i \in S} c_i \tau_i \le \sum_{i \notin S} c_i \tau_i$, then it continues to hold after the update $c_i' = c_i(1 - \tau_i/\tau_{\max})$.*

Proof. For any set T, we have

$$\sum_{i\in T} 1 - c'_i = \sum_{i\in T}(1-c_i) + \sum_{i\in T}(c_i - c'_i) = \sum_{i\in T}(1-c_i) + \frac{1}{\tau_{\max}}\sum_{i\in T} c_i\tau_i. \tag{2}$$

Applying this for $T = S$ and $T = [n]\backslash S$ yields the lemma. □

Thus our main job is to show that $\sum_{i\in S} c_i\tau_i \le \sum_{i\notin S} c_i\tau_i$. Equivalently, we wish to show that $\sum_{i\in S} c_i\tau_i \le \frac{1}{2}\sum_{i=1}^n c_i\tau_i$. For this, the following bound is helpful:

$$\begin{aligned}
\sum_{i\in S} c_i\tau_i &= \sum_{i\in S} c_i\langle x_i - \hat{\mu}_c, v^*\rangle^2 && (3)\\
&\le \sum_{i\in S}\langle x_i - \hat{\mu}_c, v^*\rangle^2 && (4)\\
&= (1-\epsilon)n\mathbb{E}_{p^*}[\langle x_i - \hat{\mu}_c, v^*\rangle^2] && (5)\\
&= (1-\epsilon)n\cdot (v^*)^\top(\mathsf{Cov}_{p^*}[X] + (\mu - \hat{\mu}_c)(\mu-\hat{\mu}_c)^\top)(v^*) && (6)\\
&\le (1-\epsilon)n\cdot(\|\mathsf{Cov}_{p^*}[X]\| + \|\mu - \hat{\mu}_c\|_2^2). && (7)
\end{aligned}$$

Here the second-to-last step uses the fact that for any θ, $\mathbb{E}[(X-\theta)(X-\theta)^\top] = \mathsf{Cov}[X] + (\theta-\mu)(\theta-\mu)^\top$.

Next note that $\|\mathsf{Cov}_{p^*}\| \le \sigma^2$ while $\|\mu - \hat{\mu}_c\|_2^2 \le \frac{8\epsilon}{1-2\epsilon}\sigma_c^2$ by the modulus of continuity bound combined with the fact that $p^*, q(c) \in \mathcal{G}_{\mathsf{cov}}(\hat{\sigma})$ and $\mathsf{TV}(p^*, q(c)) \le \frac{\epsilon}{1-\epsilon}$. Therefore, we have

$$\sum_{i\in S} c_i\tau_i \le (1-\epsilon)\sigma^2 n + \frac{8\epsilon(1-\epsilon)}{1-2\epsilon}\hat{\sigma}_c^2 n. \tag{8}$$

On the other hand, we have

$$\sum_{i=1}^n c_i\tau_i = (\sum_{i=1}^n c_i)\|\mathsf{Cov}_{q(c)}[X]\| = (\sum_{i=1}^n c_i)\hat{\sigma}_c^2 \ge (1-2\epsilon)\hat{\sigma}_c^2 n, \tag{9}$$

where the final inequality uses the fact that we have so far removed more mass from bad points than good points and hence at most 2ϵ mass in total. Recalling that we wish to show that (8) is at most half of (9), we require that

$$(1-2\epsilon)\hat{\sigma}_c^2 \ge 2(1-\epsilon)\sigma^2 + \frac{16\epsilon(1-\epsilon)}{1-2\epsilon}\hat{\sigma}_c^2, \tag{10}$$

which upon re-arrangement yields

$$\hat{\sigma}_c^2 \ge \frac{2(1-\epsilon)(1-2\epsilon)}{1-12\epsilon+12\epsilon^2}\sigma^2 \tag{11}$$

Since $\hat{\sigma}_c^2 \ge 20\sigma^2$ whenever the algorithm does not terminate, this holds as long as $\epsilon \le \frac{1}{12}$ (then the constant in front of σ^2 is $\frac{55}{3} < 20$). This shows that ($\mathcal{I}_2$) holds throughout the algorithm.

The one remaining detail is to prove that ($\mathcal{I}_2$) implies ($\mathcal{I}_1$). We wish to show that $\mathsf{TV}(p^*, q(c)) \le \frac{\epsilon}{1-\epsilon}$. We use the following formula for TV: $\mathsf{TV}(p,q) = \int \max(q(x) - p(x), 0)dx$. Let β be such that $\sum_{i=1}^n c_i = (1-\beta)n$. Then we have

$$\mathsf{TV}(p^*, q(c)) = \sum_{i\in S}\max\big(\frac{c_i}{(1-\beta)n} - \frac{1}{(1-\epsilon)n}, 0\big) + \sum_{i\notin S}\frac{c_i}{(1-\beta)n}. \tag{12}$$

If $\beta \le \epsilon$, then the first sum is zero while the second sum is at most $\frac{\epsilon}{1-\beta} \le \frac{\epsilon}{1-\epsilon}$. If on the other hand $\beta > \epsilon$, we will instead use the equality obtained by swapping p and q, which yields

$$\begin{aligned}
\mathsf{TV}(p^*, q(c)) &= \sum_{i\in S}\max\big(\frac{1}{(1-\epsilon)n} - \frac{c_i}{(1-\beta)n}, 0\big) && (13)\\
&= \frac{1}{(1-\epsilon)(1-\beta)n}\sum_{i\in S}\max((1-\beta)(1-c_i) + (\epsilon-\beta)c_i, 0). && (14)
\end{aligned}$$

Since $(\epsilon - \beta)c_i \leq 0$ and $\sum_{i \in S}(1 - c_i) \leq \epsilon n$, this yields a bound of $\frac{(1-\beta)\epsilon}{(1-\epsilon)(1-\beta)} = \frac{\epsilon}{1-\epsilon}$. We thus obtain the desired bound no matter the value of β, so $\mathsf{TV}(p^*, q(c)) \leq \frac{\epsilon}{1-\epsilon}$ whenever ($\mathcal{I}_2$) holds. This completes the proof. $\square$