CS291A Introduction to Differential Privacy **Fall 2021**

Lecture 10: Objective Perturbation (October 27)

Lecturer: Yu-Xiang Wang *Scribe: Rachel Redberg*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

Recap: Last Lecture

Posterior sampling mechanism
Posterior sampling (connected to Bayesian learning, i.e. starting with a prior and updating it) gives us privacy almost for free – it satisfies differentially privacy without having to make any big changes to the algorithm.

Statistical Jargons

- Data space $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y} = \mathcal{Z}$
  - This data representation with features x and labels y is common in supervised learning. We assume each data point (x_i, y_i) is drawn i.i.d. from some unknown distribution $\mathcal{D}$ supported on the feature and label spaces. By just calling the dataspace $\mathcal{Z}$ we can also generalize this framework to unsupervised learning.
- Hypothesis class $\mathcal{H}$
  - Abstract space that describes a family of classifiers (problem of learning is to find the best hypothesis h^* in the hypothesis class).
- Loss function $\ell(h, (x, y))$
  - Measures how well the hypothesis h matches the data (x, y), e.g. 0-1 loss measures whether how many mistakes a classifier makes.
- Risk $R(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(h, (x, y))]$
  - Typically the goal of learning is to minimize the risk, in other words find $h^* = \arg\min_{h\in\mathcal{H}} R(h)$. But we can't really do this in practice because we only have finitely many data points n.
- Empirical risk $\hat{R}_n(h) = \frac{1}{n}\sum_{i=1}^{n} \ell(h, (x_i, y_i))$
  - "Hat" denotes it's a statistical estimate of the risk, in this case an average over the samples. The weaker but more practical goal of learning is to minimize the empirical risk. Performance is measured by bounding the excess risk: $R(\mathcal{A}(Data)) - R(h^*) \leq \epsilon$. Excess risk is a measure of how much more risk your algorithm is suffering compared to that of the optimal h^*.
- Realizable setting: $R(h^*) = 0$, can find an h^* that classifies all data points perfectly.
- Agnostic setting doesn't make this assumption.

Learning with Differential Privacy: *Finite* Hypothesis Class

When the hypothesis class is finite, the utility theorem for the exponential mechanism (EM) gives us a good bound on the excess risk $R(\hat{h}_{\texttt{EM}}) - R(h^*)$ of the exponential mechanism:

$$\text{Excess risk} \leq \sqrt{\frac{\log(|\mathcal{H}|/\beta)}{n}} + \frac{\log(|\mathcal{H}|/\beta)}{n\epsilon} \quad (\textbf{Agnostic})$$

$$\text{Excess risk} \leq \frac{\log(|\mathcal{H}|/\beta)}{n} + \frac{\log(|\mathcal{H}|/\beta)}{n\epsilon} \quad (\textbf{Realizable})$$

Note that the utility function of the exponential function is the negative of the empirical risk. We can use the uniform convergence theorem to connect the results bounding the empirical risk of the exponential mechanism, $\hat{R}(\hat{h}_{\texttt{EM}}) - \min_{h \in \mathcal{H}} \hat{R}(h)$, with the results on the minimum risk of ERM: $R(\hat{h}_{\texttt{ERM}}) - R(h^*)$.

Some things to observe:

- The first term is the minimum excess risk that we need to incur even without privacy requirements; we derived this bound on the statistical error of ERM in Lecture 9 with a Hoeffding-based analysis.
- The second term comes from adding noise via the exponential mechanism satisfying ϵ-DP.
- β describes the probability that the bound on utility/excess risk fails to hold, and is close to 0.
- For reasonable choices of ϵ (e.g. $\epsilon > \frac{1}{\sqrt{n}}$), the second term (additional risk coming from differential privacy) goes to 0 as $n \to \infty$ faster than the first term (minimum excess risk coming from statistical error) does, meaning that asymptotically, requiring differential privacy does not hurt the performance of the model!

Learning with Differential Privacy: *Continuous* Hypothesis Class (Bounded VC Dimension)

VC dimension characterizes the sample complexity of a learning problem, and is used to measure the "size" of a hypothesis class even when the hypothesis class is infinite – roughly speaking, VC dimension of $\mathcal{H} \approx \log |\mathcal{H}|$.

Unfortunately, no ϵ-DP learner with finite ϵ can learn even basic problems (e.g. VC dimension = 1) for continuous hypothesis classes.

The problem of learning a threshold function has VC dimension 1. Below, we'll sketch a "packing lower bound" argument showing that no ϵ-DP algorithm $\mathcal{A}$ with finite ϵ can learn the VC class – i.e., there is no DP algorithm $\mathcal{A}$ whose performance (measured by excess risk) is any better than trivial.

Problem Set Up

- Data space $\mathcal{X} = [0, 1]$ is a line segment on $\mathbb{R}$.
- Label space $\mathcal{Y} = \{0, 1\}$ is binary.
- The hypothesis class $\mathcal{H}$ is the family of threshold functions $h \in [0, 1]$ given by $h \in \mathcal{H} : h(x) = \mathbb{1}(x \geq h)$.

Problem Goal Show for any finite $\epsilon < \infty$, there's no ϵ-DP algorithm $\mathcal{A}$ such that for all datasets Z, $\mathbb{E}[R(\mathcal{A}(\mathrm{Z})) - R(h^*)] < 0.45$. (This is in the realizable setting, so $R(h^*) = 0$.)

Proof Idea Define distributions $D_1, D_2, \ldots, D_k$ with disjoint support such that $\mathcal{A}$ is "successful" on $D_2, \ldots, D_k$ and then show this implies that $\mathcal{A}$ "fails" on D_1.

Define "success" as: for $Z_i \sim D_i$, $\mathbb{E}[\hat{R}(\mathcal{A}(Z_i))] < 0.45$. (Note the expectation is over the coin flips of $\mathcal{A}$.)

Proof Steps Consider datasets of size n.

- Divide $\mathcal{X} = [0, 1]$ into k small bins each with width $\eta = e^{-\epsilon n}$. Therefore $k = e^{\epsilon n}$ so that $k\eta = 1$.
- Define each distribution D_i with support in bin i. In particular, if the midpoint of bin i is h_i, then each D_i is a uniform distribution over $[h_i - \frac{1}{3}\eta, h_i - \frac{1}{3}\eta]$ where data points in $[h_i - \frac{1}{3}\eta, h_i]$ are classified as "-" ($y = 0$) and data points in $[h_i, h_i + \frac{1}{3}\eta]$ are classified as "+" ($y = 1$).
- Consider $Z_i \sim D_i^n$, i.e. Z_i is n datapoints sampled i.i.d. from D_i. For algorithm $\mathcal{A}$ to be "successful" on D_i, we want $\mathbb{E}[\hat{R}(\mathcal{A}(Z_i))] \leq 0.45$.
- Since each Z_i is sampled from a uniform distribution D_i, we expect on average that $\approx \frac{1}{2}$ of its datapoints are in $[h_i - \frac{1}{3}\eta, h_i]$ and classified as "-", and the other half are in $[h_i, h_i + \frac{1}{3}\eta]$. Therefore if algorithm $\mathcal{A}$ outputs a threshold function outside of either of these regions (meaning all points will be classified as "-" or all points will be classified as "+"), on average about half of the points will be misclassified.
- Assume $\mathcal{A}$ "succeeds" on $D_2, \ldots, D_k$. Then $\mathbb{E}[\hat{R}(\mathcal{A}(Z_i))] \leq 0.45$.
  This means $\Pr\left[\mathcal{A}(Z_i) \in \left[h_i - \frac{1}{3}\eta, h_i + \frac{1}{3}\eta\right]\right] > 0.9$ for $i = 2, \ldots, k$, because otherwise with probability > 0.1, the threshold function outputted by $\mathcal{A}$ would misclassify half of the data points on average and the expected error would be greater than $0.9 \times 0.5 = 0.45$.
- If we can show $\Pr\left[\mathcal{A}(Z_1) \notin [h_i - \frac{1}{3}\eta, h_i + \frac{1}{3}\eta]\right] > 0.9$, then we can think of it as with probability at least 0.9, the threshold function learned by $\mathcal{A}$ will on average classify half of the data points incorrectly. So we'd then have $\mathbb{E}[\hat{R}(\mathcal{A}(Z_i))] > 0.9 \times 0.5 = 0.45$ and we would be done.
- We can do show the above by

  (1) observing that $\{x \in \mathcal{X} : x \notin [h_1 - \frac{1}{3}\eta, h_1 + \frac{1}{3}\eta]\} \supset \bigcup_{i=2}^{k} \left[h_i - \frac{1}{3}\eta, h_i + \frac{1}{3}\eta\right]$

  (2) observing that for $i \in \{2, 3, \ldots, k\}$, Z_1 and Z_i are disjoint and therefore differ by n data points. We can then apply group privacy.

  (3) our assumption that $\mathcal{A}$ "succeeds" on $D_2, \ldots, D_k$ and our choice of $k = e^{\epsilon n}$.

  Then

$$\Pr\left[\mathcal{A}(Z_1) \notin \left[h_1 - \frac{1}{3}\eta, h_1 + \frac{1}{3}\eta\right]\right] \geq \sum_{i=2}^{k} \Pr\left[\mathcal{A}(Z_1) \in \left[h_i - \frac{1}{3}\eta, h_i + \frac{1}{3}\eta\right]\right] \tag{1}$$

$$\geq \sum_{i=2}^{k} e^{-\epsilon n} \Pr\left[\mathcal{A}(Z_i) \in \left[h_i - \frac{1}{3}\eta, h_i + \frac{1}{3}\eta\right]\right] \tag{2}$$

$$\geq e^{-\epsilon n} \times 0.9 = k e^{-\epsilon n} \times 0.9 = 0.9, \tag{3}$$

  meaning $\mathbb{E}[\hat{R}(\mathcal{A}(Z_1))] > 0.45$.

Convex Optimization Basics

Why do we care about convexity? Convex functions define a familiy of computationally efficient optimization problems with a unique minimizer and a way to tie local information (gradient at optimal solution is 0) to global information (what is the $\arg\min$).

Definition A function f is *convex* iff for all $x, y, 0 \leq \alpha \leq 1$, we have $f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y)$.

If f' exists, we can use an alternative definition of convexity based on first-order conditions:

A function f is *convex* iff for all x, y, we have $f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle$.

Strongly convex A function f is *strongly convex* iff for all x, y, we have

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\lambda}{2}||y - x||_2^2.$$

First-order optimality condition states that at the minimizer x^* of f and nowhere else, the gradient of f is 0:

$$x^* \in \operatorname*{argmin}_x f(x) \text{ iff } \nabla f(x^*) = 0.$$

Lipschitz constant of a function f is L-Lipschitz if for all x, y, $f(x) - f(y) \leq L||x - y||_2$.

Smoothness constant of a function f is β-smooth if for all x, y, $f(y) \leq f(x) + \langle y - x, \nabla f(x) \rangle + \frac{\beta}{2}||y - x||_2^2$.

Note the connection between the definitions for β-smoothness and λ-strongly convex – β-smoothness says there is a quadratic upper bound on f and λ-strongly convex says there is a quadratic lower bound on f.

Output Perturbation

Idea: if we calculate the global sensitivity of the minimizer, we can achieve (ϵ, δ)-DP just by applying the standard Laplace or Gaussian mechanisms to add noise to the output.

However, we need to add regularization $\frac{\lambda}{2}||\theta||_2^2$ or else the sensitivity could be unbounded and won't satisfy ϵ-DP for any finite ϵ!

Output Perturbation:
Sample $b \sim \mathcal{N}(0, \sigma^2 I_d)$.
Output $\hat{\theta}^P := \operatorname*{argmin}_\theta \left(L(\theta) + \frac{\lambda}{2}||\theta||_2^2 \right) + b$.

Objective Perturbation

Outputs the minimizer of a perturbed objective function by adding linear noise to the objective.

Objective Perturbation (`ObjPert`):
Sample $b \sim \mathcal{N}(0, \sigma^2 I_d)$.
Output $\hat{\theta}^P := \operatorname*{argmin}_\theta \left(L(\theta) + \frac{\lambda}{2}||\theta||_2^2 + b^T\theta \right)$.

In essence: output perturbation minimizes then perturbs, objective perturbation perturbs then minimizes.

Privacy Analysis of Output Perturbation *without regularization*

Let's first naïvely try to analyze an *un-regularized* version of output perturbation, given by:

> ***Output Perturbation (without regularization)***:
> Sample $b \sim \mathcal{N}(0, \sigma^2 I_d)$.
> Output $\hat{\theta}^P := \underset{\theta}{\text{argmin}}\, L(\theta) + b$.

We want to calculate the global L_2-sensitivity $\max_{x,x'} ||\hat{\theta}^P(x) - \hat{\theta}^P(x')||_2$ of the above.

Let x, x' differ by one individual's datapoint according to the add/remove definition of DP. We'll assume w.l.o.g. that x' contains an additional datapoint, and write

$$L(\theta) = \sum_{i=1}^{n} \ell_i(\theta) \text{ when the dataset is } x,$$
$$L(\theta) = \sum_{i=1}^{n} \ell_i(\theta) + \ell(\theta) \text{ when the dataset is } x'.$$

Define

$$\theta^* = \underset{\theta}{\text{argmin}} \sum_{i=1}^{n} \ell_i(\theta),$$
$$\tilde{\theta}^* = \underset{\theta}{\text{argmin}} \sum_{i=1}^{n} \ell_i(\theta) + \ell(\theta).$$

Therefore θ^* is the minimizer when the dataset is x, and $\tilde{\theta}^*$ is the minimizer when the dataset is x'.

First-order optimality conditions then tell us that

$$\sum_{i=1}^{n} \nabla \ell_i(\theta^*) = 0,$$
$$\sum_{i=1}^{n} \nabla \ell_i(\tilde{\theta}^*) + \nabla \ell(\tilde{\theta}^*) = 0.$$

We can then do a Taylor expansion around θ^*. By Taylor's theorem, we know there exists $v \in [\theta^*, \tilde{\theta}^*]$ (on the line segment between θ^* and $\tilde{\theta}^*$) such that

$$\sum_{i=1}^{n} \nabla \ell_i(\tilde{\theta}^*) = \underbrace{\sum_{i=1}^{n} \nabla \ell_i(\theta^*)}_{=0} + \sum_{i=1}^{n} \nabla^2 \ell_i(v) \cdot (\tilde{\theta}^* - \theta^*).$$

Then setting vector $b := \sum_{i=1}^{n} \nabla \ell_i(\tilde{\theta}^*)$, matrix $A := \sum_{i=1}^{n} \nabla^2 \ell_i(v)$, we have the equation $A \cdot (\tilde{\theta}^* - \theta^*) = b$ and can solve for $(\tilde{\theta}^* - \theta^*) = A^{-1} b$. But we haven't made any assumptions about the loss function and so A could very well be nearly singular or not invertible at all – meaning that the difference between θ^* and $\tilde{\theta}^*$ could be arbitrarily large, and the global sensitivity is unbounded!

Let's try making some additional assumptions (i.e. adding regularization so we know the objective function is λ-strongly convex) so that we can bound the global sensitivity.

Privacy Analysis of Output Perturbation *(this time, with bounded sensitivity!)*

We'll now consider the regularized objective function:

$$\begin{aligned} J(\theta) &= \sum_{i=1}^{n} \ell_i(\theta) + \frac{\lambda}{2}||\theta||_2^2 \\ &= L(\theta) + \frac{\lambda}{2}||\theta||_2^2. \end{aligned}$$

Note that $J(\theta)$ is now λ-strongly convex due to the regularization. We will also assume that $\ell(\theta)$ is L-Lipschitz.

The minimizers when the dataset is x and when the dataset is x' are given by

$$\begin{aligned} f(x) = \theta_\lambda^* &= \operatorname*{argmin}_\theta \sum_{i=1}^{n} \ell_i(\theta) + \frac{\lambda}{2}||\theta||_2^2. \\ f(x') = \tilde{\theta}_\lambda^* &= \operatorname*{argmin}_\theta \sum_{i=1}^{n} \ell_i(\theta) + \ell(\theta) + \frac{\lambda}{2}||\theta||_2^2. \end{aligned}$$

Now that we have added regularization, the function we are minimizing is λ-strongly convex. So applying the definition of strong convexity, we get

$$L(\tilde{\theta}_\lambda^*) + \frac{\lambda}{2}||\tilde{\theta}_\lambda^*||_2^2 \geq L(\theta_\lambda^*) + \frac{\lambda}{2}||\theta_\lambda^*||_2^2 + \langle \underbrace{\nabla J(\theta_\lambda^*)}_{=\vec{0}}, \tilde{\theta}_\lambda^* - \theta_\lambda^* \rangle + \frac{\lambda}{2}||\tilde{\theta}_\lambda^* - \theta_\lambda^*||_2^2$$

when the dataset is x. When the dataset is x', we have:

$$L(\theta_\lambda^*) + \ell(\theta_\lambda^*) + \frac{\lambda}{2}||\theta_\lambda^*||_2^2 \geq L(\tilde{\theta}_\lambda^*) + \ell(\tilde{\theta}_\lambda^*) + \frac{\lambda}{2}||\tilde{\theta}_\lambda^*||_2^2 + \langle \underbrace{\nabla J(\tilde{\theta}_\lambda^*) + \nabla \ell(\tilde{\theta}_\lambda^*)}_{=\vec{0}}, \tilde{\theta}_\lambda^* - \theta_\lambda^* \rangle + \frac{\lambda}{2}||\tilde{\theta}_\lambda^* - \theta_\lambda^*||_2^2$$

After adding up each side of these two inequalities above, cancelling out terms, and applying Lipschitz-smoothness:

$$\lambda||\theta_\lambda^* - \tilde{\theta}_\lambda^*||_2^2 \leq \ell(\theta_\lambda^*) - \ell(\tilde{\theta}_\lambda^*) \leq L||\theta_\lambda^* - \tilde{\theta}_\lambda^*||_2^2.$$

So

$$||\theta_\lambda^* - \tilde{\theta}_\lambda^*||_2^2 \leq \frac{L}{\lambda}.$$

Since this applies to all neighboring datasets, the above bound gives us the global sensitivity Δ_2 of output perturbation. Now we can apply the Gaussian mechanism by calibrating noise to Δ_2 and adding it to the optimal solution θ_λ^*.

Utility Analysis of Output Perturbation

To measure how well we can minimize the loss function under output perturbation, we will define the utility as $L(\hat{\theta}^P) - L(\theta^*)$, where $\hat{\theta}^P$ is the differentially private output and θ^* the true minimizer.

The two main sources of error are

1. Regularization – adding the $\frac{\lambda}{2}||\theta||_2^2$ term means that θ^*_λ (minimizer of the regularized objective) differs from θ^* (the minimizer without regularization).
2. Noise added due to DP.

We can decompose the expression for utility as follows:

$$L(\hat{\theta}^P) - L(\theta^*) = L(\hat{\theta}^P) - L(\theta^*_\lambda) + L(\theta^*_\lambda) - L(\theta^*),$$

where $\hat{\theta}^P = \theta^*_\lambda + \mathcal{N}(0, \sigma^2 I_d)$.

We've assumed that $\ell(\theta)$ is β-smooth, so $L(\theta) + \frac{\lambda}{2}||\theta||_2^2 = \sum_{i=1}^{n} \ell_i(\theta) + \frac{\lambda}{2}||\theta||_2^2$ is $n\beta + \lambda$-smooth and we can state the following:

$$L(\hat{\theta}^P) + \frac{\lambda}{2}||\hat{\theta}^P||_2^2 \leq L(\theta^*_\lambda) + \frac{\lambda}{2}||\theta^*_\lambda||_2^2 + \frac{n\beta + \lambda}{2}||\hat{\theta}^P - \theta^*||_2^2.$$

Note that $\hat{\theta}^P - \hat{\theta}^* = \mathcal{N}(0, \sigma^2 I_d)$ (the noise). Since we add i.i.d. Gaussian noise to each coordinate, we can get a high-probability bound by applying a sub-Gaussian tail bound to each coordinate and then taking a union bound over the d coordinates.

In particular, for $k \in \{1, \ldots, d\}$, $(\hat{\theta}^P - \theta^*_\lambda)_i = \mathcal{N}(0, \sigma^2) \leq \sigma\sqrt{2\log(d/\rho)}$ with probability $1 - \frac{\rho}{d}$. A union bound tells us that $\Pr\left[\bigcup_{i=1}^d (\hat{\theta}^P - \theta^*)_i > t\right] \leq \sum_{i=1}^d \Pr\left[(\hat{\theta}^P - \theta^*)_i > t\right]$. So with probability $1 - \rho$, we have that $||\hat{\theta}^P - \theta^*_\lambda||_2 \leq \sigma\sqrt{d}\sqrt{2\log(d/\rho)}$, and so

$$||\hat{\theta}^P - \theta^*_\lambda||_2^2 \leq 2d\sigma^2 \log(d/\rho).$$

Then rearranging terms and applying the above bound, we get

$$L(\hat{\theta}^P) - L(\theta^*_\lambda) \leq \frac{\lambda}{2}||\theta^*_\lambda||_2^2 - \frac{\lambda}{2}||\hat{\theta}^P||_2^2 + \frac{n\beta + \lambda}{2} \cdot 2d\sigma^2 \log(d/\rho).$$

Then by our choice of $\sigma = \frac{\Delta_2}{\epsilon}\sqrt{2\log(1.25/\delta)} = \frac{L}{\lambda\epsilon}\sqrt{2\log(1.25/\delta)}$, we have

$$L(\hat{\theta}^P) - L(\theta^*_\lambda) \leq \frac{\lambda}{2}||\theta^*_\lambda||_2^2 - \frac{\lambda}{2}||\hat{\theta}^P||_2^2 + 2(n\beta + \lambda) \cdot d\frac{L^2}{\lambda^2\epsilon^2}\log(1.25/\delta)\log(d/\rho).$$

Now we can choose λ optimally depending on whether or not $\lambda \leq n\beta$. Typically $n\beta > \lambda$ except for very smooth (almost linear) loss functions.

We could also relax the β-smoothness assumption to a Lipschitz assumption, with slightly weaker utility guarantees.

Privacy Analysis of Objective Perturbation

`ObjPert` satisfies (ϵ, δ)-DP for $\delta > 0$ with

$$\epsilon \approx d\log\left(1 + \frac{\beta}{\lambda}\right) + \frac{L^2}{2\sigma^2} + O\left(\frac{L}{\sigma}\sqrt{2\log\left(\frac{1}{\delta}\right)}\right).$$

L is a constant such that $||\nabla\ell(\cdot)||_2 \leq L$ everywhere, β is a bound on the maximum eigenvalue of the Hessian such that $\nabla^2\ell(\cdot) \prec \beta I_d$, σ is the scale of the noise added, λ is the level of regularization, and d is the dimension of the data points.

The above bound can be obtained by calculating the privacy loss random variable using a bijection between the noise vector b and the output $\hat{\theta}^P$. We can do a "change-of-variables" defined by the Jacobian matrix of the mapping between b and $\hat{\theta}^P$ to directly calculate the probability density function of $\hat{\theta}^P$, then tidy things up with some algebra and an application of the matrix determinant lemma for rank-1 updates.

We use $D_{\pm z}$ to mean the dataset obtained by adding or removing z. For completeness, here is the formal proof of the privacy analysis:

Lemma 10.1 ("Change-of-variables" for density functions)**.** *Let $g : \mathbb{R}^d \to \mathbb{R}^d$ be a bijective and differentiable function, and let X, Y be continuous random variables in $\mathbb{R}^d$ related by the transformation $Y = g(X)$. Then the probability density of Y is*

$$f_Y(y) = f_X(g^{-1}(y)) \left| det \left[\frac{\partial g^{-1}(y)}{\partial y} \right] \right|,$$

with $\left[\frac{\partial g^{-1}(y)}{\partial y} \right]$ denoting the $d \times d$ Jacobian matrix of the mapping $X = g^{-1}(Y)$.

First-order conditions tell us that

$$b(\hat{\theta}^P; D) = -\left(\nabla \hat{\mathcal{L}}(\hat{\theta}^P; D) + \nabla r(\hat{\theta}^P) + \lambda \hat{\theta}^P \right). \tag{10.1}$$

Then taking the gradient of the noise vector, we have

$$\nabla b(\hat{\theta}^P; D) = -\left(\nabla^2 \hat{\mathcal{L}}(\hat{\theta}^P; D) + \nabla^2 r(\hat{\theta}^P) + \lambda I_d \right). \tag{10.2}$$

Let $b \sim \mathcal{N}(0, \sigma^2 I_d)$, and denote $\nu(\cdot)$ as the probability density function of the normal distribution: i.e., the density at b is $\nu(b; \sigma) \propto e^{-\frac{||b||_2^2}{2\sigma^2}}$. Then since the objective function $J(\theta; D)$ is strictly convex in θ (implying that the mapping between $\hat{\theta}^P$ and b is bijective and monotonic), by Lemma 10.1 we can write

$$\begin{aligned}
\log \frac{\Pr\big(\mathcal{A}(D) = \hat{\theta}^P\big)}{\Pr\big(\mathcal{A}(D_{\pm z}) = \hat{\theta}^P\big)} &= \log \frac{\left|\det\Big(\nabla b(\hat{\theta}^P; D)\Big)\right|}{\left|\det\Big(\nabla b(\hat{\theta}^P; D_{\pm z})\Big)\right|} \frac{\nu(b(\hat{\theta}^P; D); \sigma)}{\nu(b(\hat{\theta}^P; D_{\pm z}); \sigma)} \\
&= \log \frac{\left|\det\Big(\nabla b(\hat{\theta}^P; D)\Big)\right|}{\left|\det\Big(\nabla b(\hat{\theta}^P; D_{\pm z})\Big)\right|} + \log \frac{e^{-\frac{1}{2\sigma^2}||b(\hat{\theta}^P; D))||_2^2}}{e^{-\frac{1}{2\sigma^2}||b(\hat{\theta}^P; D_{\pm z})||_2^2}} \\
&= \underbrace{\log \frac{\left|\det\Big(\nabla b(\hat{\theta}^P; D)\Big)\right|}{\left|\det\Big(\nabla b(\hat{\theta}^P; D_{\pm z})\Big)\right|}}_{(*)} + \underbrace{\frac{1}{2\sigma^2}\Big(||b(\hat{\theta}^P; D_{\pm z})||_2^2 - ||b(\hat{\theta}^P; D))||_2^2\Big)}_{(**)}.
\end{aligned}$$

Dealing first with the term (*), we observe that $\nabla b(\hat{\theta}^P; D_{\pm z}) = \nabla b(\hat{\theta}^P; D) \mp \nabla^2 \ell(\hat{\theta}^P; z)$. The notation "$\mp$" means to subtract if $z \notin D$, and add if $z \in D$. Using the eigendecomposition $\nabla^2 \ell(\hat{\theta}^P; z) = \sum_{k=1}^d \lambda_k u_k u_k^T$ and recursively applying the matrix determinant lemma, we have

$$\begin{aligned}
\Big|\det\big(\nabla b(\hat{\theta}^P; D_{\pm z})\big)\Big| &= \Big|\det\Big(\nabla b(\hat{\theta}^P; D) \mp \nabla^2 \ell(\hat{\theta}^P; z)\Big)\Big| \\
&= \Big|\det\Big(\nabla b(\hat{\theta}^P; D) \mp \sum_{k=1}^{d} \lambda_k u_k u_k^T\Big)\Big| \\
&= \Big|\det\Big(\nabla b(\hat{\theta}^P; D) \mp \sum_{k=1}^{d-1} \lambda_k u_k u_k^T \mp \lambda_d u_d u_d^T\Big)\Big| \\
&= \Big|\det\Big(\nabla b(\hat{\theta}^P; D) \mp \sum_{k=1}^{d-1} \lambda_k u_k u_k^T\Big)\Big|\Big(1 \mp \lambda_d u_d^T \big(\nabla b(\hat{\theta}^P; D) \mp \sum_{k=1}^{d-1} \lambda_k u_k u_k^T\big)^{-1} u_d\Big) \\
&= \ldots \\
&= \Big|\det\Big(\nabla b(\hat{\theta}^P; D)\Big)\Big| \prod_{j=1}^{d} \big(1 \mp \mu_j\big),
\end{aligned}$$

where $\mu_j = \lambda_j u_j^T \Big(\nabla b(\hat{\theta}^P; D) \mp \sum_{k=1}^{j-1} \lambda_k u_k u_k^T\Big)^{-1} u_j$. Therefore,

$$\begin{aligned}
(*) &= \log \frac{\Big|\det\big(\nabla b(\hat{\theta}^P; D)\big)\Big|}{\Big|\det\big(\nabla b(\hat{\theta}^P; D_{\pm z})\big)\Big|} \\
&= \log \frac{\Big|\det\big(\nabla b(\hat{\theta}^P; D)\big)\Big|}{\Big|\det\Big(\nabla b(\hat{\theta}^P; D)\Big)\Big| \prod_{j=1}^{d} \big(1 \mp \mu_j\big)} \\
&= \log \frac{1}{\prod_{j=1}^{d} \big(1 \mp \mu_j\big)} \\
&= -\log \prod_{j=1}^{d} \big(1 \mp \mu_j\big).
\end{aligned}$$

We'll handle the second term (**) next. We have that

$$\begin{aligned}
(**) &= \frac{1}{2\sigma^2}\Big(||b(\hat{\theta}^P; D_{\pm z})||_2^2 - ||b(\hat{\theta}^P; D)||_2^2\Big) \\
&= \frac{1}{2\sigma^2}\Big[\mp \nabla \ell(\hat{\theta}^P; z)\Big]\Big[2b(\hat{\theta}^P; D) \mp \nabla \ell(\hat{\theta}^P; z)\Big] \\
&= \pm \frac{1}{\sigma^2}\big[\nabla J(\hat{\theta}^P; D)^T \nabla \ell(\hat{\theta}^P; z)\big] + \frac{1}{2\sigma^2}||\nabla \ell(\hat{\theta}^P; z)||_2^2.
\end{aligned}$$

The rest of the proof follows by adding (*) and (**) and using the following bounds for each term (for GLMs):

- $\log(1 + f''(\cdot)\mu(\cdot)) \leq \log(1 + \frac{\beta}{\lambda})$.
- $||\nabla \ell(\cdot)||_2 \leq L$.
- $\nabla \ell(\cdot)^T b(\cdot) \leq \sigma L \sqrt{2 \log(2/\delta)}$.