Applications: Statistics & Machine Learning

Theo Diamandis

January 24, 2023

In this lecture, we will look at a variety of problems coming from statistics and machine learning through convex optimization. This framework allows us to incorporate prior information that traditional machine learning methods cannot handle. On homework, we've already seen cases where this provides a much better estimator. This lecture largely follows [BV04, Ch. 7].

1 Maximum Likelihood (ML) Estimation

In this section, we will examine the approximation problem, introduced in a previous lecture, through the lens of ML estimation. We consider a family of probability distributions on $\mathbf{R}^m$ parameterized by a vector $x \in \mathbf{R}^n$ and with density p_x. For a fixed $y \in \mathbf{R}^m$, the *likelihood function* is $p_x(y)$. For convenience, we work with its logarithm, the *log-likelihood function*, denoted ℓ:

$$\ell(x) = \log p_x(y).$$

Maximum Likelihood Estimate. Consider the problem of estimating the parameter vector x after observing a sample y. Perhaps we also have some prior information, $x \in C$. Then the *maximum likelihood* estimate of x is a solution to the optimization problem

$$\begin{array}{ll} \text{maximize} & \ell(x) = \log p_x(y) \\ \text{subjectto} & x \in C. \end{array}$$

Note that y is problem data and not a variable here. This method is widely used and is a useful common parent of many other problems. If p is log-concave, then this problem is a convex optimization problem (which is the case for many distributions in practice.) Sometimes, you'll have to make an additional change of variables in practice (*e.g.,* using the inverse of the covariance matrix Σ^{-1} instead of Σ as a variable.)

Linear measurements with IID noise. Consider the linear measurement model

$$y = a_i^T x + v_i, \quad i = 1, \dots, m.$$

The ML estimate is the solution to the problem

$$\text{maximize} \quad \ell(x) = \sum_{i=1}^{m} \log p(y_i - a_i^T x).$$

We assume p is log-concave, so that this problem is a convex optimization problem. Note that it has the same form as the approximation problem we saw earlier. In fact, several penalty functions are readily derived from common noise distributions.

- **Gaussian Noise.** When $v_i \sim \mathcal{N}(\mu, \sigma)$, the ML estimate is $\hat{x} = \text{argmin}_x \|Ax - y\|_2^2$
- **Laplacian Noise.** When v_i is Laplacian, the ML estimate is $\hat{x} = \text{argmin}_x \|Ax - y\|_1$
- **Uniform Noise.** When v_i is uniform on $[-a, a]$, the ML estimate is any x such that $\|Ax - y\|_\infty \leq a$.

In fact, the penalty function problem,

$$\text{maximize} \quad \sum_{i=1}^{m} \phi(y_i - a_i^T x),$$

can be interpreted as the ML estimate under a linear measurement model with noise density

$$p(z) = C \cdot e^{-\phi(z)},$$

where $C^{-1} = \int e^{-\phi(u)} du$. This formulation allows us to interpret the penalty problem statistically. For example, if ϕ increases rapidly for large values of u, this means that the noise distribution has small tails.

Logistic regression. Consider a random variable $z \in \{0, 1\}$. We have data $\{(y_i, z_i)\}_{i=1}^m$ where $y_i \in \mathbf{R}^n$ is a feature vector and $z_i \in \{0, 1\}$. Binary logistic regression is the ML estimation problem for the distribution

$$p_{x,w}(z = 1;\, y) = \frac{\exp(x^T y + w)}{1 + \exp(x^T y + w)}.$$

Here, $x \in \mathbf{R}^n$ and $w \in \mathbf{R}$ are the variables (parameters of the distribution) and $y \in \mathbf{R}^n$ is the feature vector. The log-likelihood function is then

$$\ell(x, w) = \sum_{i=1}^{m} \left(z_i(x^T y + w) - \log(1 + \exp(x^T y + w)) \right),$$

which is concave.

2 Sparse Regression

We revisit using the ℓ_1 norm as a convex approximation to the cardinality function, denoted $\mathbf{card}(x)$, which is the number of nonzero entries in x. This function appears in many problems but unfortunately is not convex (it is quasiconcave though). Examples include finding sparse design (*e.g.,* a filter or circuit with minimum number of hardware components), handling fixed transaction costs (*e.g.,* for logistics planning), and estimating with outliers (*e.g.,* allow k arbitrary violations of the model). One popular applications is the sparse regression problem

$$\begin{array}{ll} \text{minimize} & \mathbf{card}(x) \\ \text{subjectto} & \|Ax - y\|_2^2 \leq \delta, \end{array}$$

where δ is a chosen tolerance.

The ℓ_1 heuristic. The most common approach to tackle cardinality problems is to replace $\mathbf{card}(x)$ with $\gamma\|x\|_1$ or add the regularization term $\gamma\|x\|_1$ to the objective, where γ is a parameter used to achieve a desired sparsity. Reformulating the sparse regression problem, we have

$$\begin{array}{ll} \text{minimize} & \|x\|_1 \\ \text{subject to} & \|Ax - y\|_2^2 \leq \delta, \end{array}$$

Other variants include the LASSO problem

$$\begin{array}{ll} \text{minimize} & \|Ax - y\|_2^2 \\ \text{subject to} & \|x\|_1 \leq \beta, \end{array}$$

or the basis pursuit denoising problem

$$\text{minimize} \quad \|Ax - y\|_2^2 + \gamma\|x\|_1.$$

This heuristic is quite good; in fact, under some assumptions, it will reconstruct x exactly with high probability.

Iterated weighted ℓ_1 heuristic. In fact, we can do somewhat better that the ℓ_1 heuristic to minimize cardinality. Consider the following procedure:

- Let $w = \mathbf{1}$
- Repeat
  - Replace $\mathbf{card}(x)$ with $\|\,\mathbf{diag}\,(w)\,x\|_1$ in the objective
  - Solve the problem. Let the solution be x^k
  - Let $w_i = 1/(\varepsilon + |x_i^k|)$

This procedure actually uses the approximation

$$\mathbf{card}(z) \approx \log(1 + z/\varepsilon) \approx \log(1 + z^0/\varepsilon) + \frac{z - z^0}{\varepsilon + z^0}$$

and solves the problem by linearizing this nonconvex function at each iterate. (This fact is important if you have other terms in the objective or **card** in the constraints, since then you should not ignore the constants and positive scaling factors.)

Solution polishing. After finding a solution $\hat{x}$ with required sparsity via the ℓ_1 heuristic, it is a good idea to 'polish' the solution. Fix the sparsity pattern of x to be that of $\hat{x}$ and re-solve the optimization problem with this sparsity pattern to obtain a final heuristic solution $x^\star$.

3 Support vector machine

The support vector machine (SVM) attempts to find a linear discriminator that approximately separates two sets of points $X = \{x_1, \ldots, x_N\}$ and $Y = \{y_1, \ldots, y_M\}$. In other words, we want to find a and b such that

$$a^T x_i + b > 0 \qquad \text{and} \qquad a^T y_i + b < 0.$$

Since the problem is homogenous in a and b, we instead work with

$$a^T x_i + b \geq 1 \qquad \text{and} \qquad a^T y_i + b \leq -1.$$

Usually these two sets are not exactly separable, so we have some number of errors. We solve the optimization problem

$$\begin{array}{ll} \text{minimize} & \|a\|_2 + \gamma \mathbf{1}^T(u + v)) \\ \text{subject to} & a^T x_i + b \geq 1 - u_i, \quad i = 1, \ldots, N \\ & a^T y_i + b \leq -1 + v_i, \quad i = 1, \ldots, M \\ & u \geq 0, \quad v \geq 0. \end{array}$$

The objective trades off between maximizing the margin $2/\|a\|_2$ and minimizing the classification error $\mathbf{1}^T(u+v)$. Of course, we can look at other discriminators, such as the quadratic discriminant function

$$x_i^T P x_i + q^T x_i + r \geq 1 \qquad \text{and} \qquad y_i^T P y_i + q^T y_i + r \leq -1.$$

Note that these lead to linear constraints in the variables, which are P, q, and r here.

References

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex optimization.* Cambridge university press, 2004.