(Stochastic) Gradient Descent

Empirical Risk Functional $R_{\text{emp}}[f] = \frac{1}{n} \sum_{i=1}^{m} c(x_i, y_i, f(x_i))$

**Idea 1**

Minimize $R_{\text{emp}}[f]$ by performing gradient descent. This leads to

$$f \rightarrow f - \frac{\Lambda}{m} \sum_{i=1}^{m} \partial f_c(x_i, y_i, f(x_i))$$

**Problem**

This may be expensive. If the observations are similar, this is very wasteful.

**Idea 2**

Minimize $R_{\text{emp}}[f]$ by performing stochastic gradient descent over the individual terms under the sum.

Stochastic Gradient $f \rightarrow f - \frac{\Lambda}{m} \partial c(x_i, y_i, f(x_i))$

Linear Model $w \rightarrow w - \frac{\Lambda}{m} \partial c(x_i, y_i, f(x_i))$

---

Perceptron Algorithm for Squared Loss

**argument:** Training sample, $\{x_1, \ldots, x_n\} \subset X$, $\{y_1, \ldots, y_n\} \subset \{\pm 1\}$, $\eta$

**returns:** Weight vector $w$ and threshold $b$.

**function Perceptron**$(X, Y, \eta)$

initialize $w, b = 0$

repeat

for all $i$ from $i = 1, \ldots, m$

Compute $f(x_i) = \langle \sum_{t=1}^{i} \alpha_t \phi(x_i), \phi(x_i) \rangle + b$

Update $w, b$ according to $w' = w + \eta \alpha t \phi(x_i)$ and $b' = b + \eta \alpha t$

endfor

until for all $1 \leq i \leq m$ we have $g(x_i) = y_i$

return $f : x \mapsto \langle w, \Phi(x) \rangle + b$

---

Perceptron Algorithm for Huber’s Loss

**argument:** Training sample, $\{x_1, \ldots, x_n\} \subset X$, $\{y_1, \ldots, y_n\} \subset \{\pm 1\}$, $\eta$

**returns:** Weight vector $w$ and threshold $b$.

**function Perceptron**$(X, Y, \eta)$

initialize $w, b = 0$

repeat

for all $i$ from $i = 1, \ldots, m$

Compute $f(x_i) = \langle \sum_{t=1}^{i} \alpha t \phi(x_i), \phi(x_i) \rangle + b$

Update $w, b$ according to $w' = w + \eta \alpha t \phi(x_i)$ and $b' = b + \eta \alpha t$

where $\alpha t = \begin{cases} \frac{1}{\sigma}(y_i - f(x_i)) & \text{for } |y_i - f(x_i)| \leq \sigma \\ sgn(y_i - f(x_i)) & \text{otherwise} \end{cases}$

endfor

until for all $1 \leq i \leq m$ we have $g(x_i) = y_i$

return $f : x \mapsto \langle w, \Phi(x) \rangle + b$

---

Learning Rate

**Classification**

For classification, the absolute value of $f$ does not matter. So we need not adjust the learning rate.

**Regression**

The absolute value of $f$ is crucial, so we have to get $\eta$ right.

- Large $\eta$: we get quick initial convergence to the target but large fluctuations remain (stochastic gradient can be very noisy).
- Small $\eta$: slow initial convergence to the target but we have a much better quality estimate in the later stages.

**Trick**

Make $\eta$ a variable of the time. One can show that $\eta(t) = O(1/t)$ is optimal in many cases. This yields quick initial convergence and low fluctuations later.

**Warning**

If $f$ is fluctuating, choosing $\eta$ too small will not be useful.
Maximum Likelihood and Noise Models

**Basic Idea**
We assume that the observations $y_i$ are derived from $f(x_i)$ by adding noise, i.e. $y_i = f(x_i) + \xi_i$ where $\xi_i$ is a random variable with density $p(\xi)$. This also means that once we know the type of noise we are dealing with, we may compute conditional densities $p(y|x)$ under the model assumptions.

**Likelihood** $p(Y|f, X) = p((y_1 - f(x_1)), \ldots, (y_m, f(x_m)))$
We make the assumption of iid data (to keep the equations simple). This leads to the likelihood
\[ \mathcal{L} = \prod_{i=1}^{m} p(y_i - f(x_i)) \]

**Caveat**
The estimates we obtain are only as good as our initial assumptions regarding the type of function expansion and noise. This means that we may not take $p(Y|X)$ at book value.

---

Log-Likelihood and Loss Function

**Idea**
Log likelihood and loss function look suspiciously similar, maybe we can find a link . . . . For simplicity we assume that the that is generated iid.

**Comparison**
\[ -\mathcal{L}[f] = \sum_{i=1}^{m} \log p(y_i - f(x_i)) \]
\[ R_{\text{emp}}[f] = \frac{1}{m} \sum_{i=1}^{m} c(x_i, y_i, f(x_i)) \]

**Idea**
The two terms differ only by a scaling constant which is irrelevant for minimization purposes. So match up the terms.
\[ c(x, y, f(x)) \equiv -\log p(y_i - f(x_i)) \]
\[ p(y_i|f(x_i)) \equiv \exp(-c(x_i, y_i, f(x_i))) \]

---

Density and Loss

<table>
<thead>
<tr>
<th>loss function $c(\xi)$</th>
<th>density model $p(\xi)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_{\text{insensitive}}(\xi)$</td>
<td>$\frac{1}{2</td>
</tr>
<tr>
<td>$c_{\text{Laplace}}(\xi)$</td>
<td>$\frac{1}{2} \exp(-</td>
</tr>
<tr>
<td>$c_{\text{Gaussian}}(\xi)$</td>
<td>$\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\xi^2}{2}\right)$</td>
</tr>
<tr>
<td>$c_{\text{Huber’s}}$ robust loss</td>
<td>$\begin{cases} \frac{1}{\sigma}(\xi)^2 &amp; \text{if }</td>
</tr>
</tbody>
</table>

**Function Expansion**
We use a linear model (as in the previous lecture) $f_1, \ldots, f_r$, such that
\[ f(x) = \sum_{i=1}^{n} \alpha_i f_i(x) \]

**Additive Noise**
Assume Gaussian noise $\xi$ which corrupts the measurements such that we observe $y$ rather than $f(x)$, i.e. $y = f(x) + \xi$. We write $\xi \sim N(0, \sigma)$ in order to state that
\[ p(\xi) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}\xi^2\right) \]

**Density Model**
From above we know that $p(y|x, \alpha, \sigma)$ is given by
\[ p(y|x, \alpha, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y - f(x))^2\right) \]
A Worked-Through Example, Part II

Likelihood

Under the assumption of iid data, the likelihood of observing $Y = \{y_1, \ldots, y_m\}$, given $X = \{x_1, \ldots, x_m\}$ can be found as

$$p(Y|X, \alpha, \sigma) = \prod_{i=1}^{m} p(y_i|x_i, \alpha, \sigma)$$

Log Likelihood

$$\mathcal{L} = \sum_{i=1}^{m} \log p(y_i|x_i, \alpha, \sigma)$$

$$= \sum_{i=1}^{m} \log \frac{1}{\sqrt{2\pi \sigma^2}} \exp \left( -\frac{1}{2\sigma^2} (y_i - f(x_i))^2 \right)$$

$$= -\frac{m}{2} \log(2\pi \sigma^2) - \frac{1}{\sigma^2} \sum_{i=1}^{m} (y_i - f(x_i))^2$$

A Worked-Through Example, Part III

Optimality Criterion

We need a maximum with respect to the parameters $\alpha, \sigma$. The conditions $\partial_{\alpha} \mathcal{L} = 0$ and $\partial_{\sigma} \mathcal{L} = 0$ are necessary for this purpose.

Optimality in $\alpha$

$$\partial_{\alpha} \mathcal{L} = \partial_{\alpha} \frac{1}{2\sigma^2} \|y - F\alpha\|^2 = \frac{1}{\sigma^2} (F^T F \alpha - F^T y) = 0$$

Here we defined (as before) $F_i = f_j(x_i)$. It leads to the standard least mean squares solution $\alpha = (F^T F)^{-1} F^T y$.

Optimality in $\sigma$

$$\partial_{\sigma} \mathcal{L} = \frac{m}{\sigma} - \frac{1}{\sigma^2} \sum_{i=1}^{m} (y_i - f(x_i))^2 = 0$$

Likewise this leads to $\sigma = \frac{1}{m} \sum_{i=1}^{m} (y_i - f(x_i))^2$ which is empirical variance given by the model on the training set.

When Things go wrong with ML

No fine-grained prior knowledge

All functions we optimize over are treated as equally likely.

Not possible to check assumptions

- Our ML model works if the assumptions are correct. However, it breaks if they are not all satisfied. And it is hard to test them.
- Difficult to integrate alternative estimates.
- Confidence bounds for estimates.

High dimensional estimates break

- Overly confident estimates
- Overfitting
- Likelihood diverges: assume $y_i = f(x_i)$. In this case we would estimate $\sigma = 0$ as the empirical variance. This in turn leads to $\mathcal{L} \to \infty$.

Regularization

Problem

The space of the solutions for $f$ is too large if we admit all possible solutions in, say, the span of $f_1, \ldots, f_r$. Moreover we want to rank the solutions.

Idea

Restrict the possible solutions to the set $\Omega[f] \leq c$ where $\Omega[f]$ is some convex function

$$\Omega[f] = \sum_{i=1}^{n} |\alpha_i| (\ell_1 \text{ Regularization})$$

$$\Omega[f] = \frac{1}{2} \sum_{i=1}^{n} \alpha_i^2 (\ell_2 \text{ Regularization})$$

$$\Omega[f] = \frac{1}{2} \alpha^T M\alpha \text{ where } M \text{ is a positive semidefinite matrix}$$
Regularized Risk Functional

Problem
Restricting $f$ to the subset $\Omega[f] \leq c$ will solve the problem but the optimization problems are sometimes rather difficult to solve.

Idea
Trade off the size of $\Omega[f]$ with respect to $R_{\text{size}}[f]$ and minimize the sum of these two terms.

Definition
For some $\lambda > 0$, also referred to as the regularization constant, the regularized risk functional is given by

$$R_{\text{reg}}[f] = R_{\text{size}} + \lambda \Omega[f] = \frac{1}{m} \sum_{i=1}^{m} c(x_i, y_i, f(x_i)) + \lambda \Omega[f]$$

This is the central quantity in most learning settings. Note that $R_{\text{reg}}[f]$ is convex, provided $R_{\text{size}}[f]$ and $\Omega[f]$ are.

Example: Adding to the Diagonal

Quadratic Loss $c(x, y, f(x)) = \frac{1}{2}(y - f(x))^2$

Linear Model $f(x) = \sum_{i=1}^{n} \alpha_i f_i(x)$

$\ell_2$ Regularizer $\Omega[f] = \sum_{i=1}^{n} \alpha_i^2$

Regularized Risk Functional

$$R_{\text{reg}}[f] = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{2} (y_i - f(x_i))^2 + \frac{\lambda}{2} \sum_{i=1}^{n} \alpha_i^2 = \frac{1}{2m} \|y - Fa\|^2 + \frac{\lambda}{2} \|\alpha\|^2$$

Optimality Conditions

$$\partial_{\alpha} R_{\text{reg}}[f] = \frac{1}{m} (F^T y + F^T Fa) + \lambda \alpha = 0$$

This is the same as when we added $\varepsilon$ to the main diagonal to invert matrices or improve their condition.

A Practical Example

- Training Set
- Regression for $\lambda = 0.1$
- Regression for $\lambda = 1$
- Regression for $\lambda = 10$