Optimization for ML

Last lecture:

Logistic regression is a Discriminative model, it models

$$P(y|\mathbf{x}_i)$$

Functional form of logistic regression

$$P(Y=0|X) = \frac{1}{1+\exp(w_0 + \sum_i w_i X_i)}$$

How to train logistic regression?

Learn the parameters $w_0, w_1, \ldots, w_d$ from training data:

$$\{(X^{(j)}, Y^{(j)})\}_{j=1}^{n} \qquad X^{(j)} = (X_1^{(j)}, \ldots, X_d^{(j)})$$

Last lecture:

Learn the parameters $w_0, w_1, \ldots, w_d$ from training data:

$$\{(X^{(j)}, Y^{(j)})\}_{j=1}^{n} \qquad X^{(j)} = (X_1^{(j)}, \ldots, X_d^{(j)})$$

In such a way as to maximize *conditional likelihood estimates*:

$$\hat{\mathbf{w}}_{MCLE} = \arg\max_{\mathbf{w}} \prod_{j=1}^{n} P(Y^{(j)} \mid X^{(j)}, \mathbf{w})$$

Maximizing log-likelihood:

$$\begin{aligned} l(\mathbf{w}) &\equiv \ln \prod_j P(y^j | \mathbf{x}^j, \mathbf{w}) \\ &= \sum_j \left[y^j (w_0 + \sum_i^d w_i x_i^j) - \ln(1 + exp(w_0 + \sum_i^d w_i x_i^j)) \right] \end{aligned}$$

…or minimizing negative log-likelihood

Optimization

$$
\begin{aligned}
l(\mathbf{w}) &\equiv \ln \prod_j P(y^j|\mathbf{x}^j, \mathbf{w}) \\
&= \sum_j \left[y^j(w_0 + \sum_i^d w_i x_i^j) - \ln(1 + exp(w_0 + \sum_i^d w_i x_i^j)) \right]
\end{aligned}
$$

No closed-form solution to maximize the log-likelihood

What is optimization?

Finding (one or more) maximizer/minimizer of a function subject to constraints

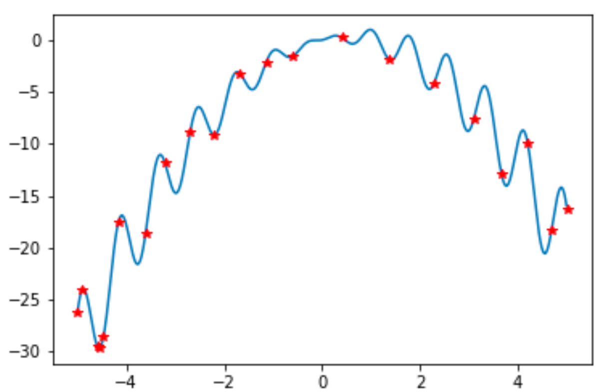
$$
\begin{aligned}
&\arg\min_x f_0(x) \\
&\quad \text{s.t.} f_i(x) \leq 0, i = \{1, \ldots, k\} \\
&\quad\quad h_j(x) = 0, j = \{1, \ldots l\}
\end{aligned}
$$

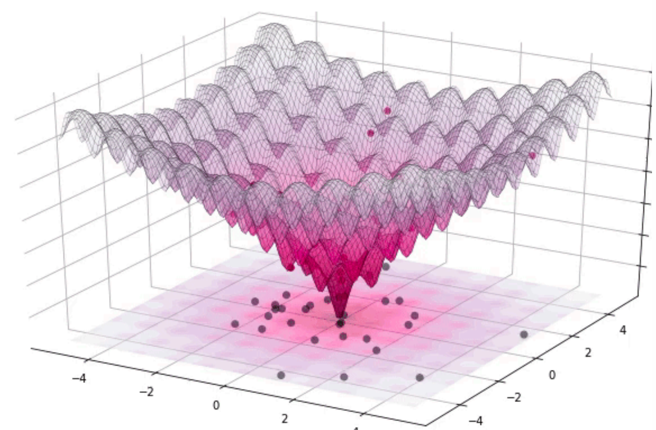
Optimization

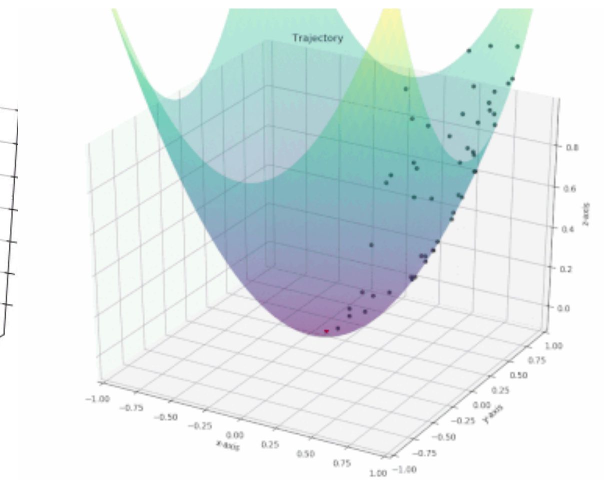
Most of the machine learning problems are, in the end, optimization problems

$$\widehat{\mathbf{w}}_{MCLE} = \arg\max_{\mathbf{w}} \prod_{j=1}^{n} P(Y^{(j)} \mid X^{(j)}, \mathbf{w})$$

Optimization in general is difficult







Optimization and concave/convex functions

$l(w)$ is a concave function

$-l(w)$ is a convex function

Nice property: strictly concave/convex function has a unique maximum/minimum

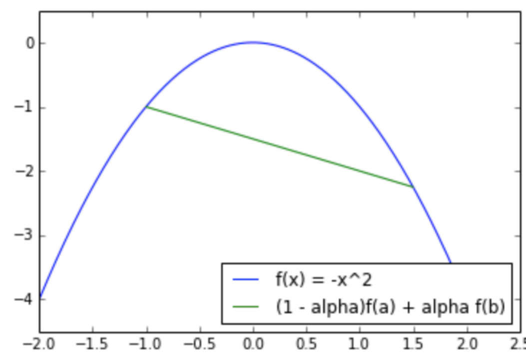
Why $l(w)$ concave?

*Recall: a function **f** is concave if*

$$f((1-\alpha)a + \alpha b) \geq (1-\alpha)f(a) + \alpha f(b) \quad \forall a, b, \quad 0 \leq \alpha \leq 1$$

*Or **f** is convex if*

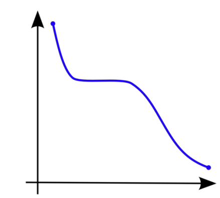
$$f((1-\alpha)a + \alpha b) \leq (1-\alpha)f(a) + \alpha f(b) \quad \forall a, b, \quad 0 \leq \alpha \leq 1$$



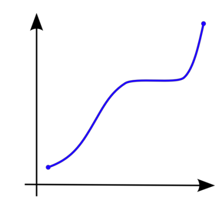
Optimization and concave/convex functions

Operations that preserve convexity:

- -f is concave if and only if f is convex
- Nonnegative weighted sums:
  - If $\alpha_1, \ldots, \alpha_n \geq 0$ and $f_1, \ldots, f_n$ are all convex then $\alpha_1 f_1 + \ldots + \alpha_n f_n$ is convex
- If f and g are convex functions and g is non-decreasing over a univariate domain, then $h(x) = g(f(x))$ is convex
- If f is concave and g is convex and non-increasing over a univariate domain, then $h(x) = g(f(x))$ is convex



non-increasing function



non-decreasing function

Optimization and concave/convex functions

Examples:

- The functions $f(x) = -x^2$ and $g(x) = \sqrt{x}$ are concave

- The function $f(x) = \log(x)$ is concave on its domain

- Any affine function $f(x) = ax + b$ is both concave and convex but neither strictly-concave nor strictly-convex

- For f being convex, the function $g(x) = e^{f(x)}$ is convex because e^x is convex and monotonically increasing

Optimization and concave/convex functions

Why $l(w)$ concave?

- Use the definition (lot of math!)
- In one dimension: If the second derivative is negative on an interval then f is concave
- In higher dimensions: matrix of second derivatives (Hessian) is negative semi definite.
- Check this: http://qwone.com/~jason/writing/convexLR.pdf

Or: *sum of concave functions is a concave function!*

$$= \sum_j \left[\underbrace{y^j (w_0 + \sum_i^d w_i x_i^j)}_{\text{Affine function (concave)}} - \underbrace{\ln(1 + exp(w_0 + \sum_i^d w_i x_i^j))}_{\text{negative log-sum-exp of an affine function}} \right]$$

Optimization and concave/convex functions

Linear/affine functions:

$$f(x) = b^T x + c.$$

Quadratic functions:

$$f(x) = \frac{1}{2} x^T A x + b^T x + c$$

Norms (like ℓ_1 or ℓ_2 for regularization):

$$\|\alpha x + (1-\alpha)y\| \leq \|\alpha x\| + \|(1-\alpha)y\| = \alpha \|x\| + (1-\alpha)\|y\| .$$

Composition with an affine function $f(Ax+b)$:

$$\begin{aligned} f(A(\alpha x + (1-\alpha)y) + b) &= f(\alpha(Ax+b) + (1-\alpha)(Ay+b)) \\ &\leq \alpha f(Ax+b) + (1-\alpha) f(Ay+b) \end{aligned}$$

Gradient ascent/descent

So: $l(w)$ is concave, now what?

- It has a unique maximum! (easy to find it)
- Maximum of concave function can be reached by gradient ascent

Gradient descent is a first-order iterative optimization algorithm for finding the minimum/maximum of a function.

For a optimization problem with a concave function ***f***:

For $k = 1$ until *stop-criteria (change < eps)*

Do: $x_{t+1} \leftarrow x_t + \eta \nabla f(x_t)$

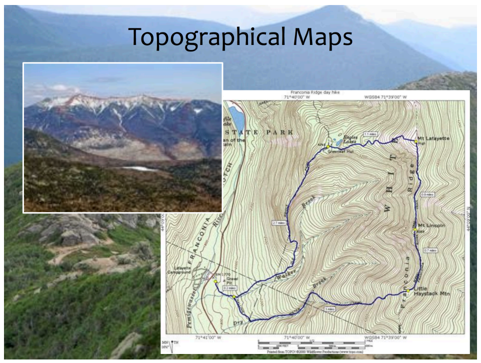
Gradient of f

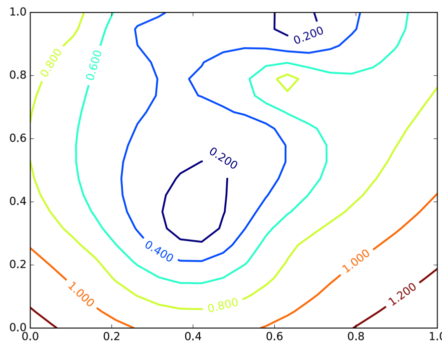
next step

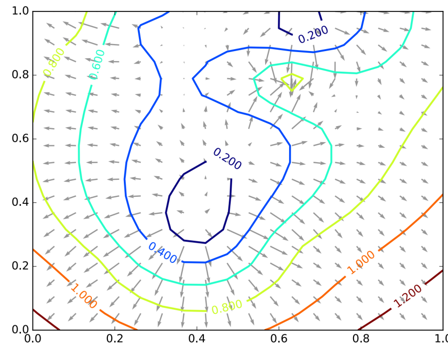
Learning rate / step size

Current step

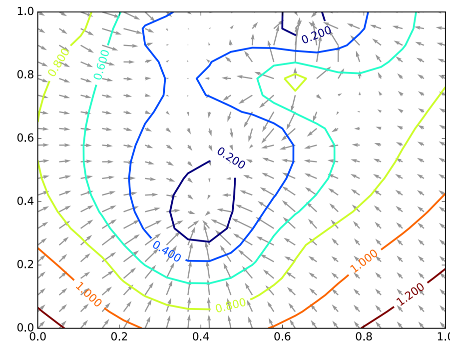
Gradient ascent/descent





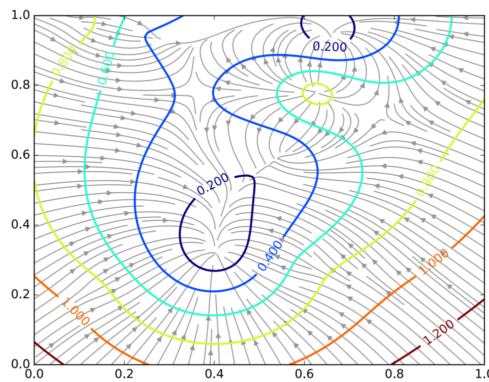


These are the **gradients** that Gradient **Ascent** would follow.



These are the **negative** gradients that Gradient **Descent** would follow.

(Negative) Gradient *Paths*



Shown are the **paths** that Gradient Descent would follow if it were making **infinitesimally small steps.**

Gradient ascent/descent for logistic regression

Gradient ascent rule for w_0:

$$w_0^{(t+1)} \leftarrow w_0^{(t)} + \eta \left.\frac{\partial l(\mathbf{w})}{\partial w_0}\right|_t$$

(Pick $w_0^{(0)}$ at random)

$$l(\mathbf{w}) = \sum_j \left[y^j (w_0 + \sum_i^d w_i x_i^j) - \ln(1 + exp(w_0 + \sum_i^d w_i x_i^j)) \right]$$

(log-likelihood)

$$\frac{\partial l(\mathbf{w})}{\partial w_0} = \sum_j \left[y^j - \frac{1}{1 + exp(w_0 + \sum_i^d w_i x_i^j)} \cdot exp(w_0 + \sum_i^d w_i x_i^j) \right]$$

(derivate wrt. w_0)

$$w_0^{(t+1)} \leftarrow w_0^{(t)} + \eta \sum_j [y^j - \hat{P}(Y^j = 1 \mid \mathbf{x}^j, \mathbf{w}^{(t)})]$$

(update rule for w_0)

For i=1,…,d,

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \sum_j x_i^j [y^j - \hat{P}(Y^j = 1 \mid \mathbf{x}^j, \mathbf{w}^{(t)})]$$

(update rule for w_i)

Gradient ascent/descent for logistic regression

- **Tips and suggestions for Gradient Descent**
  - **Plot Cost versus Time**: Plot the values of the function ***f*** calculated by the algorithm on each iteration. Performing gradient ascent increases the value of function ***f*** in each iteration. If it does not decrease, try reducing your learning rate.

  - **Learning Rate**: The learning rate value is a small real value such as 0.1, 0.001 or 0.0001. Try different values for your problem and see which works best.

  - **Rescale Inputs:** The algorithm will reach the minimum cost faster if the shape of the cost function is not skewed and distorted. You can achieved this by rescaling all of the input variables (X) to the same range, such as [0, 1] or [-1, 1].