Logistic Regression

---

1 Problem Formulation

Input space:

X⊆Rd\mathcal{X} \subseteq \mathbb{R}^dX⊆Rd

Output space:

Y={0,1}\mathcal{Y} = \{0,1\}Y={0,1}

Data distribution:
i.i.d. samples from unknown distribution.

Learning Objective:

min⁡wE[ℓ(Y,fw(X))]\min_w \mathbb{E}[\ell(Y, f_w(X))]wminE[ℓ(Y,fw(X))]

---

2 Model Specification

Hypothesis function:

P(Y=1|X=x)=σ(wTx)P(Y=1|X=x) = \sigma(w^T x)P(Y=1|X=x)=σ(wTx)

Where sigmoid:

σ(z)=11+e−z\sigma(z) = \frac{1}{1 + e^{-z}}σ(z)=1+e−z1

Parameter space:

w∈Rdw \in \mathbb{R}^dw∈Rd

Structural assumptions:

- Linear decision boundary
- Log-odds are linear in features

---

3 Loss Function

Binary Cross-Entropy:

L(y,y^)=−[ylog(y^)+(1−y)log(1−y^)]L(y,\hat{y}) = -[y\log(\hat{y}) + (1-y)\log(1-\hat{y})]L(y,y^)=−[ylog(y^)+(1−y)log(1−y^)]

Why this loss?

- Derived from Bernoulli likelihood
- Proper scoring rule

Convexity:

- Convex in www
- No closed-form solution

4️⃣ Objective Function

Empirical Risk:

J(w)=−1n∑i=1n[yilog(σ(wTxi))+(1−yi)log(1−σ(wTxi))]J(w) = -\frac{1}{n}\sum_{i=1}^{n} \left[y_i \log(\sigma(w^T x_i)) + (1-y_i)\log(1-\sigma(w^T x_i))\right]J(w)=−n1i=1∑n[yilog(σ(wTxi))+(1−yi)log(1−σ(wTxi))]

Regularized (L2):

J(w)+λ||w||2J(w) + \lambda ||w||^2J(w)+λ||w||2

5️⃣ Optimization Method

Iterative method:

- Gradient Descent
- Newton’s Method

Gradient:

∇J(w)=1nXT(σ(Xw)−y)\nabla J(w) = \frac{1}{n} X^T (\sigma(Xw) - y)∇J(w)=n1XT(σ(Xw)−y)

Convergence:

- Convex ⇒ global optimum
- Slower than linear regression

Complexity:

- O(nd)O(nd)O(nd) per iteration

---

6 Statistical Interpretation

MLE connection:
Assume:

Y|X∼Bernoulli(p)Y|X \sim Bernoulli(p)Y|X∼Bernoulli(p)

Then maximizing likelihood = minimizing cross-entropy.

Noise model: Bernoulli

Probabilistic meaning:
Outputs:

P(Y=1|X)P(Y=1|X)P(Y=1|X)

---

7 Regularization & Generalization

Bias–Variance:

- High C (low λ) → overfitting
- High λ → underfitting

Capacity control:

- L1 (feature selection)
- L2 (weight shrinkage)

---

8 Theoretical Properties

- Convex loss
- Unique global minimum
- Consistent classifier (under assumptions)

9️⃣ Computational Complexity

Training:

- O(ndk)O(ndk)O(ndk) (k iterations)

Inference:

- O(d)O(d)O(d)

Memory:

- O(nd)O(nd)O(nd)

🔟 Limitations

- Cannot model non-linear boundaries (without feature engineering)
- Sensitive to multicollinearity
- Requires proper scaling
- Fails if classes perfectly separable (without regularization)