Day 6: Expectation and Variance — Mean and Spread of Random Variables

Quantifying Central Tendency and Dispersion in AI Systems

1. The Fundamental Question

> *How do we mathematically capture both the typical behavior and the uncertainty around that behavior for random variables?*

Expectation and variance provide the fundamental statistical measures that quantify central tendency and dispersion, forming the bedrock of loss functions, optimization objectives, and uncertainty quantification in AI.

2. Mathematical Foundations

2.1. Expectation (Expected Value)

For discrete random variables:

$$E[X] = \sum_{x \in \mathcal{X}} x \cdot p_X(x)$$

For continuous random variables:

$$E[X] = \int_{-\infty}^{\infty} x \cdot f_X(x) dx$$

2.2. Variance and Standard Deviation

$$\text{Var}(X) = E[(X - E[X])^2] = E[X^2] - (E[X])^2$$

$$\sigma_X = \sqrt{\text{Var}(X)}$$

2.3. Properties and Rules

Key Mathematical Properties

- **Linearity of Expectation:** $E[aX + bY + c] = aE[X] + bE[Y] + c$
- **Variance Scaling:** $\text{Var}(aX + b) = a^2\text{Var}(X)$
- **Expectation of Functions:** $E[g(X)] = \sum_x g(x)p_X(x)$ (discrete) or $\int g(x)f_X(x)dx$ (continuous)
- **Total Variance:** $\text{Var}(X) = E[\text{Var}(X|Y)] + \text{Var}(E[X|Y])$ (Law of Total Variance)

3. Comprehensive Examples and Analysis

3.1. Discrete Case: Biased Coin Toss

Bernoulli Distribution Analysis

- $X \sim \text{Bernoulli}(p)$ with $p = 0.6$
- PMF: $P(X = 1) = 0.6$, $P(X = 0) = 0.4$
- Applications: Binary classification, A/B testing, success/failure outcomes

Step-by-Step Calculations:

$$
\begin{aligned}
E[X] &= 1 \cdot 0.6 + 0 \cdot 0.4 = 0.6 \\
E[X^2] &= 1^2 \cdot 0.6 + 0^2 \cdot 0.4 = 0.6 \\
\text{Var}(X) &= E[X^2] - (E[X])^2 = 0.6 - (0.6)^2 = 0.6 - 0.36 = 0.24 \\
\sigma_X &= \sqrt{0.24} \approx 0.4899
\end{aligned}
$$

Interpretation: The expected success rate is 60%, with a standard deviation of approximately 49% indicating high relative variability.

3.2. Continuous Case: Normal Distribution

Normal Distribution Moments

- $Y \sim \mathcal{N}(\mu, \sigma^2)$
- PDF: $f_Y(y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y-\mu)^2}{2\sigma^2}}$
- Applications: Error modeling, natural phenomena, central limit theorem applications

Moment Calculations:

$$\begin{aligned} E[Y] &= \mu \\ \text{Var}(Y) &= \sigma^2 \\ E[Y^2] &= \mu^2 + \sigma^2 \\ \text{Skewness} &= 0 \quad \text{(symmetric distribution)} \\ \text{Kurtosis} &= 3 \quad \text{(mesokurtic)} \end{aligned}$$

3.3. Complex Example: Dice Roll with Reward Function

Nonlinear Transformation

- X: Fair six-sided die outcome
- Reward function: $g(X) = X^2$ (squared reward)
- Calculate $E[g(X)]$ and $\text{Var}(g(X))$

Calculations:

$$\begin{aligned} E[X] &= 3.5, \quad \text{Var}(X) = 2.9167 \\ E[g(X)] &= E[X^2] = \frac{1}{6}(1 + 4 + 9 + 16 + 25 + 36) = \frac{91}{6} \approx 15.1667 \\ E[g(X)^2] &= E[X^4] = \frac{1}{6}(1 + 16 + 81 + 256 + 625 + 1296) = \frac{2275}{6} \approx 379.1667 \\ \text{Var}(g(X)) &= E[g(X)^2] - (E[g(X)])^2 \approx 379.1667 - (15.1667)^2 \approx 149.3056 \end{aligned}$$

4. Why Expectation and Variance are Fundamental to AI

Core Applications in Machine Learning

- **Loss Functions:** MSE = Expected squared error, Cross-entropy = Expected negative log-likelihood
- **Optimization:** Gradient descent minimizes expected loss
- **Regularization:** Variance reduction through methods like dropout, weight decay
- **Uncertainty Quantification:** Predictive variance for confidence intervals
- **Model Selection:** Bias-variance tradeoff guiding model complexity

5. Real-World AI Applications

5.1. Mean Squared Error in Regression

- **Objective:** Minimize $E[(y - \hat{y})^2]$
- **Decomposition:** MSE = Bias^2 + Variance + Irreducible Error
- **Performance:** Root MSE typically 10-30% of target variable range in well-tuned models
- **Optimization:** Closed-form solution for linear models, gradient-based for neural networks

5.2. Variance Reduction in Deep Learning

- **Batch Normalization:** Reduces internal covariate shift, stabilizing training
- **Dropout:** Approximate Bayesian inference, reduces overfitting
- **Weight Initialization:** Xavier/He initialization maintains activation variance across layers
- **Ensemble Methods:** Reduce predictive variance by averaging multiple models

5.3. Reinforcement Learning

- **Value Functions:** $V(s) = E[\sum \gamma^t r_t]$ - expected cumulative reward
- **Policy Gradient:** Maximize expected return through policy parameters
- **Risk-Sensitive RL:** Consider variance of returns for safer policies
- **Exploration-Exploitation:** Balance expected reward and uncertainty

6. Implementation in Modern AI Frameworks

6.1. PyTorch Implementation

```
import torch
import torch.distributions as dist

# Basic expectation and variance calculations
x = torch.tensor([1.0, 2.0, 3.0, 4.0, 5.0])
probabilities = torch.tensor([0.1, 0.2, 0.4, 0.2, 0.1])

# Manual calculation
expectation = torch.sum(x * probabilities)
variance = torch.sum(probabilities * (x - expectation)**2)

print(f"E[X] = {expectation:.4f}")
print(f"Var(X) = {variance:.4f}")

# Using PyTorch distributions
normal = dist.Normal(0.0, 1.0)
samples = normal.sample((10000,))
empirical_mean = torch.mean(samples)
empirical_var = torch.var(samples)

print(f"Empirical mean: {empirical_mean:.4f}")
print(f"Empirical variance: {empirical_var:.4f}")

# Variance of model predictions
predictions = model(input_data)
prediction_variance = torch.var(predictions, dim=0)
```

6.2. Advanced Statistical Functions

```
# Law of Total Variance example
def law_of_total_variance(conditional_means, conditional_variances, weights):
    """
    Var(X) = E[Var(X|Y)] + Var(E[X|Y])
    """
    expected_conditional_variance = torch.sum(weights * conditional_variances)
    variance_of_conditional_means = torch.var(conditional_means, weights=weights)
    return expected_conditional_variance + variance_of_conditional_means

# Monte Carlo expectation
def monte_carlo_expectation(func, distribution, num_samples=10000):
    samples = distribution.sample((num_samples,))
    return torch.mean(func(samples))
```

7. Advanced Concepts and Extensions

7.1. Higher Moments and Moment Generating Functions

- **Skewness:** $\gamma_1 = E[(\frac{X-\mu}{\sigma})^3]$ - measure of asymmetry
- **Kurtosis:** $\gamma_2 = E[(\frac{X-\mu}{\sigma})^4]$ - measure of tail heaviness
- **MGF:** $M_X(t) = E[e^{tX}]$ - generates all moments
- **Cumulants:** Logarithm of MGF, useful for independent sums

7.2. Conditional Expectation and Variance

- **Conditional Expectation:** $E[X|Y]$ - random variable, function of Y
- **Tower Property:** $E[E[X|Y]] = E[X]$
- **Law of Total Variance:** $\text{Var}(X) = E[\text{Var}(X|Y)] + \text{Var}(E[X|Y])$
- **Applications:** Bayesian inference, missing data imputation, control variates

7.3. Concentration Inequalities

- **Markov's Inequality:** $P(X \geq a) \leq \frac{E[X]}{a}$ for $a > 0$
- **Chebyshev's Inequality:** $P(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}$
- **Chernoff Bound:** $P(X \geq a) \leq \min_{t>0} e^{-ta} M_X(t)$

- **Applications:** PAC learning, generalization bounds, algorithm analysis

8. Performance Analysis and Empirical Validation

Statistical Validation in AI Systems

- **Moment Matching:** Compare empirical and theoretical moments
- **Convergence Monitoring:** Track loss variance during training
- **Uncertainty Calibration:** Ensure predictive variance matches empirical error
- **Bias-Variance Decomposition:** Analyze model error sources
- **Monte Carlo Error:** Error $\propto \frac{\sigma}{\sqrt{N}}$ for N samples

9. Practical Exercises for Mastery

Hands-On Expectation and Variance Analysis

1. **Bernoulli Moments:** Derive $E[X]$, $\text{Var}(X)$, $E[X^3]$ for $X \sim \text{Bernoulli}(p)$
2. **Exponential Distribution:** Calculate moments for $X \sim \text{Exp}(\lambda)$
3. **Law of Total Variance:** Verify with a two-stage experiment
4. **Empirical Validation:** Generate samples and compare empirical vs theoretical moments
5. **Neural Network Analysis:** Track activation means and variances across layers

10. Common Pitfalls and Best Practices

Expectation and Variance Guidelines

- **Linearity Misuse:** Remember $E[g(X)] \neq g(E[X])$ for nonlinear g
- **Variance of Sum:** $\text{Var}(X+Y) = \text{Var}(X) + \text{Var}(Y) + 2\text{Cov}(X,Y)$
- **Small Sample Bias:** Use Bessel's correction: $s^2 = \frac{1}{n-1}\sum(x_i - \bar{x})^2$
- **Numerical Stability:** Use Welford's algorithm for online variance computation
- **Distribution Assumptions:** Check moment existence (e.g., Cauchy distribution has undefined mean)

11. Historical Context and Modern Impact

11.1. Historical Development

- **17th Century:** Huygens introduces concept of expected value
- **18th Century:** Bernoulli and De Moivre develop early variance concepts
- **19th Century:** Gauss and Laplace formalize method of moments
- **20th Century:** Fisher's analysis of variance (ANOVA), Markov's inequalities

11.2. Modern AI Applications

- **Deep Learning:** Batch normalization using running mean/variance estimates
- **Bayesian Methods:** Posterior expectation and variance for uncertainty
- **Reinforcement Learning:** Expected return optimization
- **Generative Models:** Matching data distribution moments

12. Key Insight: The Dual Nature of Uncertainty

Expectation and variance together provide a complete picture of random behavior in AI systems:

- **Expectation as Center:** Captures the typical or average behavior
- **Variance as Spread:** Quantifies the uncertainty or risk around that center

- **Complementary Nature:** Both are needed for complete understanding
- **Optimization Trade-offs:** Balancing bias and variance in model selection

The mathematical elegance of these concepts lies in their ability to:

- **Summarize Complexity:** Reduce complex distributions to interpretable numbers
- **Guide Decisions:** Expected value for optimization, variance for risk assessment
- **Enable Theory:** Foundation for convergence proofs and generalization bounds
- **Drive Innovation:** New algorithms based on moment matching and variance reduction

Mastering expectation and variance is not just about mathematical proficiency—it's about developing the statistical intuition needed to build robust, reliable AI systems that properly account for uncertainty in their predictions and decisions.

Next: Covariance and Correlation — Understanding Relationships Between Variables

Tomorrow we'll explore how to measure and model relationships between random variables—the mathematical foundation for feature selection, dimensionality reduction, and understanding complex dependencies in AI systems.