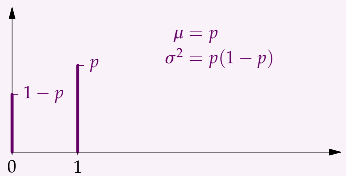
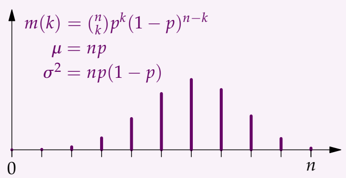
DATA 1010 · Brown University · Samuel S. Watson

Probability: Common Distributions

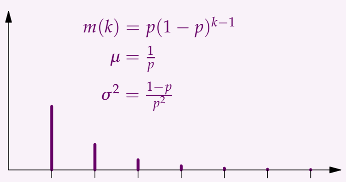
1 **Bernoulli** (Ber(p)): A weighted coin flip



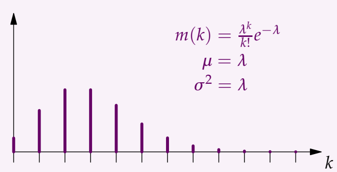
2 **Binomial** (Bin(n,p)): A sum of n independent Ber(p)'s



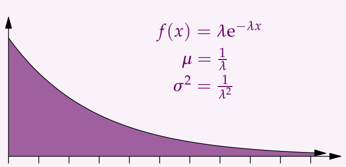
3 **Geometric** (Geom(p)): Time to first success (1) in a sequence of independent Ber(p)'s



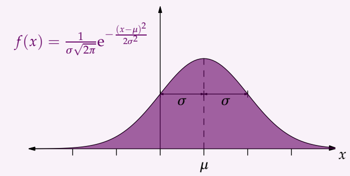
4 **Poisson distribution** (Poiss(λ)): Limit as $n \to \infty$ of Binomial($n, \frac{\lambda}{n}$)



5 **Exponential distribution** (Exp(λ)): Limit as $n \to \infty$ of distribution of $1/n$ times a Geometric(λ/n)



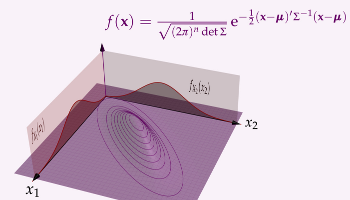
6 **Normal distribution** ($\mathcal{N}(\mu, \sigma^2)$): Limit as $n \to \infty$ of the distribution of $\frac{X_1 + X_2 + \cdots + X_n}{\sqrt{n}}$, for any independent sequence $X_1, \ldots, X_n$ of identically distributed random variables (i.i.d.) with $\mathbb{E}[X_1] = \mu$ and $\mathrm{Var}(X_1) = \sigma^2 < \infty$ (see Central Limit Theorem).



7 **Multivariate normal distribution** ($\mathcal{N}(\mathbf{0}, \Sigma)$): if $\boldsymbol{Z} = (Z_1, Z_2, \ldots, Z_n)$ is a vector of independent $\mathcal{N}(0,1)$'s, A is an $m \times n$ matrix of constants, and $\mu \in \mathbb{R}^m$, then the vector

$$\boldsymbol{X} = A\boldsymbol{Z} + \mu$$

is **multivariate normal**. The covariance matrix of $\mathbf{X}$ is $\Sigma = AA'$.



Machine Learning

1 The KDE cross-validation loss estimator is

$$J(f) = \int_{\mathbb{R}} \widehat{f}_\lambda^2 - \frac{2}{n}\sum_{i=1}^n \widehat{f}_\lambda^{(-i)}(X_i),$$

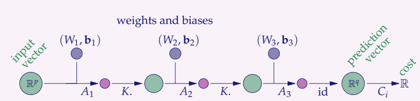
2 The logistic regression loss estimator is

$$L(r) = \sum_{i=1}^n \left[y_i \log \frac{1}{r(x_i)} + (1-y_i) \log \frac{1}{1-r(x_i)} \right],$$

3 The SVM loss estimator is

$$L(\boldsymbol{\beta}, \alpha) = \lambda |\boldsymbol{\beta}|^2 + \frac{1}{n}\sum_{i=1}^n \left[1 - y_i(\boldsymbol{\beta} \cdot \mathbf{x}_i - \alpha)\right]_+$$

4 Our neural net diagram:



Learning Standards

- [ ] **1** [SETFUN] Correctly answer questions about basic set and function terminology
- [ ] **2** [JULIA] Write Julia code to solve simple algorithmic problems using conditionals, functions, arrays, dictionaries, and iteration.
- [ ] **3** [LINALG] Use vocabulary and results from linear algebra to solve problems involving linear independence, span, and rank.
- [ ] **4** [MATALG] Use matrix algebra (including matrix transposes) to solve problems involving projection and orthogonality
- [ ] **5** [EIGEN] Apply knowledge of determinants, eigendecomposition, and singular value decomposition to data problems and other applications
- [ ] **6** [OPT] Explain the Lagrange multipliers theorem and gradient descent and discuss issues surrounding applied optimization
- [ ] **7** [MATDIFF] Differentiate matrix expressions with respect to vectors and use this technique to solve optimization problems.
- [ ] **8** [MACHARITH] Reason about 64-bit and 32-bit floating point arithmetic
- [ ] **9** [NUMERROR] Discuss the categories of numerical error and identify points of concern in application
- [ ] **10** [PRNG] Discuss basic considerations surrounding the generation of pseudorandom numbers, such as seed, period, and statistical tests
- [ ] **11** [COUNTING] Use the fundamental principle of counting and binomial coefficients to solve basic counting problems
- [ ] **12** [PROBSPACE] Explain the elements of a probability space and use probability spaces to model random experiments
- [ ] **13** [PMF] Reason about discrete random variable distributions and use properties of discrete distributions to solve problems
- [ ] **14** [PDF] Reason about continuous random variable distributions and use properties of continuous distributions to solve problems
- [ ] **15** [CONDPROB] Use the conditional probability formula to translate back and forth between branching tree diagrams and their corresponding probability spaces
- [ ] **16** [BAYES] Use Bayes' theorem and other properties of conditional probability to solve conditional probability problems
- [ ] **17** [IND] Explain independence of random variables, construct a probability space with independent random variables, and use independence to solve probability problems
- [ ] **18** [EXP] Use the definition of a random variable, the distribution of the random variable, or linearity of expectation to find the expectation of a random variable
- [ ] **19** [COV] Calculate variances and covariances, recognize high or low variance and positive or negative covariance from graphical representations of distributions, and use properties of variance and covariance to solve problems about random variable distributions
- [ ] **20** [CONDEXP] Calculate conditional expectations and apply them to expectation problems
- [ ] **21** [COMDISTD] Discuss definitions and properties of common discrete distributions (Bernoulli, binomial, geometric) and recognize circumstances under which those distributions can be expected to fit the data well
- [ ] **22** [COMDISTC] Discuss definitions and properties of common continuous distributions (Poisson, exponential, multivariate normal)
- [ ] **23** [RVINEQ] Explain inequalities involving random variable expectations (such a Chebyshev's inequality) and use them to solve problems
- [ ] **24** [CLT] State and apply the central limit theorem, and recognize when the conclusion of the central limit theorem should not be expected to hold
- [ ] **25** [KDE] Apply kernel density estimators to data problems, and explain ways of dealing with the bias-variance tradeoff in density estimation
- [ ] **26** [LR] Explain the techniques of basic linear and polynomial regression, and discuss the advantages and disadvantages relative to nonparametric methods
- [ ] **27** [QDA] Discuss the assumptions of, the estimation methods for, and facts about quadratic and linear discriminant analysis
- [ ] **28** [STATLEARN] Explain the main points of statistical learning theory (regression vs classification, loss functional, target function, learner, training and test error, overfitting, inductive bias, bias-variance tradeoff)
- [ ] **29** [NPL] Apply classification vocabulary (confusion matrix, detection rate, false alarm rate, precision, receiver operating characteristic) and the Neyman-Pearson lemma to reason about classification problems
- [ ] **30** [SVM] Describe the mathematics and intuition behind support vector machines (both hard- and soft-margin)
- [ ] **31** [LOGIST] Describe, apply, and analyze logistic regression models
- [ ] **32** [NN] Describe, apply, and analyze multi-layer perceptrons for regression and classification
- [ ] **33** [DR] Describe and interpret dimension reduction methods, including principal component analysis (concept and technical details) and t-SNE (concept only)
- [ ] **34** [R] Perform basic programming tasks in R (defining variables, generating and indexing matrices, writing functions, and reading/writing to disk)
- [ ] **35** [GGPLOT] Use `ggplot` to create data visualizations (data, aesthetics, geometries, statistics, scales, faceting)
- [ ] **36** [DPLYR] Apply the six fundamental verbs in Hadley Wickham's grammar of data manipulation (`filter`, `arrange`, `select`, `mutate`, `group_by`, `summarise`) to transform data