EECS 553 Lecture "Exam1 Review" Machine Learning (ECE) [class version]

Jeff Fessler

University of Michigan
©Laura Balzano and Jeff Fessler, UM, October 5, 2026, 10:50

F26

Exam1 logistics

© 2026 Fessler

- Exam on Mon. Oct. 12, 6-8PM (sharp!)
- Surname A-L: 1303 EECS
- Surname M-Z: 1311 EECS
- Two pages of 8.5x11" paper, both sides
- Handwritten recommended; typed OK
- No magnifying devices
- No other paper
- Closed book, closed notes
- No calculators, phones, devices, smart glasses, smart watches....
- Scope: lectures 1-9 and HW 1-4 (including non-graded problems)

Outline

© 2026 Fessler

- Prereqs: probability and matrix analysis
- Supervised learning probabilistic setting
- Empirical risk minimization
- Bayes classifier
- Plug-in classifiers
- Regression / logistic regression
- Unconstrained optimization
- Constrained optimization
- Separating hyperplanes, max-margin
- Kernel methods

Background: probability and matrix analysis

© 2026 Fessler

- Joint and conditional probability
- Expectation
- Independence
- Law of total probability
- Matrix norms
- Matrix trace
- PSD matrix properties
- SVD, eigenvalue decomposition, rank
- and more ...

Some distributions you must know (seen on HW)© 2026 Fessler

- Uniform
- Multivariate gaussian
- Bernoulli
- Categorical
- Poisson
- Negative binomial

Put each PMF or pdf on your cheat sheet if you don't have them memorized.

Supervised learning probabilistic setting

© 2026 Fessler

- Feature vector $\boldsymbol{x} \in \mathbb{R}^d$
- Label $y \in \mathcal{Y}$
  - Either $\{1, \dots, K\}$ for classification, or $\{\pm 1\}$ or $\{0, 1\}$ for binary classification, or $\mathbb{R}$ or $\mathbb{R}^k$ for regression
- Assume $(\boldsymbol{x}, y)$ form a jointly distributed ($d+1$ dimensional) random vector.
- A function $f : \mathbb{R}^d \mapsto \mathcal{Y}$ is the output of ML training
  - The classifier or regression function
- Loss is a function $L : \mathcal{Y} \times \mathcal{Y} \mapsto [0, \infty)$.
- Surrogate loss is a function that is easier to optimize, but should still lead to "good" classifier design.
- Risk:

$$R(f) \triangleq \mathbb{E}_{\boldsymbol{x}, y}\{L(y, f(\boldsymbol{x}))\}$$

Empirical risk minimization

© 2026 Fessler

- n feature vectors / data vectors $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \in \mathbb{R}^d$
- Corresponding labels $\{\boldsymbol{y}_1, \ldots \boldsymbol{y}_n\}$ with $\boldsymbol{y}_i \in \mathcal{Y}$
- A function $f : \mathbb{R}^d \mapsto \mathcal{Y}$ – the output of ML training
- Loss $L : \mathcal{Y} \times \mathcal{Y} \mapsto [0, \infty)$.
- Empirical Risk:

$$R(f) \triangleq \frac{1}{n} \sum_{i=1}^{n} L(\boldsymbol{y}_i, f(\boldsymbol{X}))$$

- Feature vector $\boldsymbol{X} \in \mathbb{R}^d$, label $Y \in \{1, \ldots, K\}$, assume $(\boldsymbol{X}, Y)$ are jointly distributed ($d+1$ dimensional)
- Function $f : \mathbb{R}^d \mapsto \{1, \ldots, K\}$ is the classifier.
- Notation:
  - $\pi_k \triangleq \mathsf{P}\{Y = k\}$
  - $g_k(\boldsymbol{x}) \triangleq$ pdf/pmf of $\boldsymbol{X}$ given $Y = k$
  - $\eta_k(\boldsymbol{x}) \triangleq \mathsf{P}\{Y = k \mid \boldsymbol{X} = \boldsymbol{x}\}$
- The best classifier depends on the distribution and the performance measure. If we use **0-1 loss**, the risk is:

$$R(f) \triangleq \mathbb{E}_{\boldsymbol{X},Y}\left\{\mathbb{I}_{\{f(\boldsymbol{X}) \neq Y\}}\right\} = \mathsf{P}\{f(\boldsymbol{X}) \neq Y\}$$

$$= \mathbb{E}_{\boldsymbol{X},Y}\left\{\mathbb{I}_{\{\mathsf{sign}(\tilde{f}(\boldsymbol{X})) \neq Y\}}\right\} \text{ (binary case with } \pm 1)$$

- The smallest risk of any classifier is called the Bayes' Risk and is denoted R_*. We know examples where R_* is zero, $\frac{1}{2}$, etc.
- If $R(f) = R_*$, f is called a Bayes' classifier.
- **Theorem:** The following classifier is a Bayes' classifier:

$$\begin{aligned} f_*(\boldsymbol{x}) &= \underset{k=1,\dots,K}{\arg\max}\ \eta_k(\boldsymbol{x}) \\ &= \underset{k=1,\dots,K}{\arg\max}\ \pi_k g_k(\boldsymbol{x}) \\ &= \underset{k=1,\dots,K}{\arg\max}\ (\log(\pi_k) + \log(g_k(\boldsymbol{x}))). \end{aligned}$$

- The latter version is often more numerically stable.
- In the two-class case, the likelihood ratio test is a Bayes' classifier.
- When $\mathcal{Y} = \{0, 1\}$ then $\eta_1(\boldsymbol{x}) = \mathrm{P}\{Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}\}$ and $f(\boldsymbol{x}) = \mathbb{I}_{\{\eta_1(\boldsymbol{x}) > 1/2\}}$.

Plug-in classifiers

© 2026 Fessler

- Linear discriminant analysis, Quadratic discriminant analysis, and Naïve Bayes have the form

$$f(\boldsymbol{x}) \triangleq \arg\max_k \widehat{\pi}_k \widehat{g}_k(\boldsymbol{x})$$

- Logistic regression has the form

$$f(\boldsymbol{x}) \triangleq \arg\max_k \widehat{\eta}_k(\boldsymbol{x})$$

- We often use maximum likelihood to estimate model parameters (Illustrated in a few HW problems.)

- LDA assumes the classes are Gaussian with shared covariance.
- QDA assumes the classes are Gaussian with different covariance.
- You should know the formulae for estimates in these classifiers and the final decision functions.
- Naïve Bayes assumes the feature distributions are independent.
- Logistic regression makes the logistic assumption on the posterior.
- Know generative v discriminative, parametric v nonparametric.

Regression

© 2026 Fessler

- Linear regression, ridge regression, and logistic regression
- Know how to solve them with:
  - Gradient descent
  - Newton's method
  - Stochastic gradient descent
- Other regularizers (LASSO, elastic net...)

An *unconstrained optimization problem* has the general form

$$\arg\min_{\boldsymbol{x} \in \mathbb{R}^p} f(\boldsymbol{x})$$

where $f : \mathbb{R}^p \mapsto \mathbb{R}$ is called the **cost function**.

- First order necessary condition for a stationary point
- Second order necessary condition for a max/min
- Definitions of convexity and smoothness
- Convex functions: $\nabla f(\boldsymbol{x}_*) = \mathbf{0}$ is both necessary *and* sufficient for $\boldsymbol{x}_*$ to be a local minimizer.

- Gradient descent
  - Consider minimizing the generic objective function $J(\boldsymbol{\theta})$
  - Initial guess $\boldsymbol{\theta}_0$, some step size(s) α
  - For $t = 0, \ldots,$ `max_iter`
    $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \alpha \nabla J(\boldsymbol{\theta}_t)$
    If convergence condition satisfied, exit
    End
- Newton's method

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha [\nabla^2 J(\boldsymbol{\theta}_t)]^{-1} \nabla J(\boldsymbol{\theta}_t)$$

- Stochastic Gradient descent
  for $J(\boldsymbol{\theta}) = \sum_{i=1}^{n} J_i(\boldsymbol{\theta})$, cycle through i

$$\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \alpha \nabla J_i(\boldsymbol{\theta}_t)$$

- Subdifferentials and subgradients

A constrained optimization problem has the form

$$\begin{aligned} \underset{\boldsymbol{u} \in \mathbb{R}^p}{\arg\min}\ & f(\boldsymbol{u}) \\ \text{s.t.}\ \ & g_i(\boldsymbol{u}) \le 0, \quad i = 1, \dots, r \\ & h_j(\boldsymbol{u}) = 0, \quad j = 1, \dots, s, \end{aligned}$$

where $f, g_1, \dots, g_r, h_1, \dots, h_s : \mathbb{R}^p \mapsto \mathbb{R}$.

- Lagrangian
- Duality properties
- KKT conditions and when they imply optimality
- Convex program and relation to KKT conditions

Separating hyperplanes, Max-Margin

© 2026 Fessler

- Definition of a hyperplane and a separating hyperplane.
- Max-margin hyperplane problem
- Soft-margin hyperplane problem

These topics were in scope for Exam1 in W26, but are not in scope for Exam1 in F26:

- (Kernel) SVM
- conditions for support vectors

Kernel methods

© 2026 Fessler

These topics were in scope for Exam1 in W26, but are not in scope for Exam1 in F26.

- Nonlinear classification or regression via nonlinear feature maps
- Kernels / kernel trick using inner products
- Kernel method for OSM, *i.e.*, the Support Vector Machine
- Kernel Nearest Neighbor Classifier
- Kernel Ridge Regression