Announcements

- Collaboration policy on Psets
- Projects
- Clarifications for Problem Set 1
The correlation question

- \( N \) values in each of two samples:
  - \( e_i = y_i - \hat{w}^T x \) the prediction error
  - \( z_i = a^T x_i \) a linear function evaluated on the training examples.

- Show that \( \sigma(e_i, z_i) = 0 \).

- Develop an intuition, before you attack the derivation: Play with these in Matlab!
  - Generate a random \( w^* \), random \( X \)
  - Compute \( X w^* \), generate and add Gaussian noise
  - Fit \( \hat{w} \), calculate \( \{e_i\} \)
  - Generate a random \( a \), calculate \( \{z_i\} \). plot them!
  - Calculate correlation.
More notation

\[ A \triangleq B \] means \( A \) is defined by \( B \) (first time \( A \) is introduced).

\[ A \equiv B \] for varying \( A \) and/or \( B \) means they are always equal.

E.g., \( f(x) \equiv 1 \) means \( f \) returns 1 regardless of the input \( x \).

\( a \sim p(a) \) random variable \( a \) is drawn from density \( p(a) \)
Review

- Uncertainty in $\hat{w}$ as an estimate of $w^*$:

  \[ \hat{w} \sim \mathcal{N}(w; w^*, \sigma^2(X^TX)^{-1}) \]

- Generalized linear regression

  \[ f(x; w) = w_0 + w_1\phi_1(x) + w_2\phi_2(x) + \ldots + w_m\phi_m(x) \]

- Multivariate Gaussians
Today

- More on Gaussians
- Introduction to classification
- Projections
- Linear discriminant analysis
Refresher on probability

- Variance of a r.v. $a$: $\sigma_a^2 = E[(a - \mu_a)^2]$, where $\mu_a = E[a]$.

- Standard deviation: $\sqrt{\sigma_a^2}$. Measures the spread around the mean.

- Generalization to two variables: covariance

  $$\text{Cov}_{a,b} \triangleq E_{p(a,b)} \left[ (a - \mu_a)(b - \mu_b) \right]$$

- Measures how the two variable deviate together from their means ("co-vary").
Correlation and covariance

- Correlation:

\[
\text{cor}(a, b) \triangleq \frac{\text{Cov}_{a,b}}{\sigma_a \sigma_b}.
\]

- \( \text{cor}(a, b) \) measures the linear relationship between \( a \) and \( b \).

- \(-1 \leq \text{cor}(a, b) \leq +1 \); \( +1 \) or \( -1 \) means \( a \) is a linear function of \( b \).
Covariance matrix

- For a random vector $\mathbf{x} = [x_1, \ldots, x_d]^T$, 
  
  $$
  \text{Cov}_x \triangleq \begin{bmatrix}
  \sigma_{x_1}^2 & \text{Cov}_{x_1, x_2} & \ldots & \text{Cov}_{x_1, x_d} \\
  \text{Cov}_{x_2, x_1} & \sigma_{x_2}^2 & \ldots & \text{Cov}_{x_2, x_d} \\
  \vdots & \vdots & \ddots & \vdots \\
  \text{Cov}_{x_d, x_1} & \text{Cov}_{x_d, x_2} & \ldots & \sigma_{x_d}^2
  \end{bmatrix}.
  $$

- Square, symmetric, non-negative main diagonal (variances $\geq 0$) 

- Under that definition, one can show:

  $$
  \text{Cov}_x = E \left[ (\mathbf{x} - \mu_x)(\mathbf{x} - \mu_x)^T \right]
  $$

  i.e. expectation of the outer product of $\mathbf{x} - \mu_x$ with itself.

- Note: so far nothing Gaussian-specific!
Covariance matrix decomposition

- Any covariance matrix $\Sigma$ can be decomposed:

$$\Sigma = R \begin{bmatrix} \lambda_1 & \cdots & \lambda_d \end{bmatrix} R^T$$

where $R$ is a rotation matrix, and $\lambda_j \geq 0$ for all $j = 1, \ldots, d$.

- Rotation in 2D:

$$R = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$
Rotation matrices

\[ \Sigma = R \begin{bmatrix} \lambda_1 & \cdots & \lambda_d \end{bmatrix} R^T \]

- Rotation matrix \( R \):
  - orthonormal: if columns are \( r_1, \ldots, r_d \), then \( r_i^T r_i = 1 \), \( r_i^T r_j = 0 \) for \( i \neq j \).
  - From here follows \( R^T = R^{-1} \) (\( R^T \) reverses the rotation produced by \( R \)).
  - Columns \( r_i \) specify the basis for the “new” (rotated) coordinate system.

- \( R \) determines the orientation of the ellipse (so called \textit{principal directions})

- The inner \( \text{diag}(\lambda_1, \ldots, \lambda_d) \) specifies the scaling along each of the principal directions.

- Interpretation of the whole product: rotate, scale, and rotate back.
Covariance and correlation for Gaussians

- Suppose (for simplicity) $\mu = 0$. What happens if we rotate the data by $R^T$?

- The new covariance matrix is just

$$
\begin{bmatrix}
\sigma^2_{x_1} & \text{Cov}_{x_1,x_2} & \cdots & \text{Cov}_{x_1,x_d} \\
\text{Cov}_{x_2,x_1} & \sigma^2_{x_2} & \cdots & \text{Cov}_{x_2,x_d} \\
\vdots & \vdots & \ddots & \vdots \\
\text{Cov}_{x_d,x_1} & \text{Cov}_{x_d,x_2} & \cdots & \sigma^2_{x_d}
\end{bmatrix}
= 
\begin{bmatrix}
\lambda_1 & \cdots & \\
\cdots & \ddots & \\
\cdots & \cdots & \lambda_d
\end{bmatrix}
$$

- The components of $x$ are now uncorrelated (covariances are zero). This is known as whitening transformation.
  - For Gaussians, this also means they are independent.
  - Not true for all distributions!
Classification versus regression

• Formally: just like in regression, we want to learn a mapping from $\mathcal{X}$ to $\mathcal{Y}$, but $\mathcal{Y}$ is discrete and finite.

• One approach is to (naïvely) ignore that $\mathcal{Y}$ is such.

• Regression on the *indicator matrix*:
  
  – Code the possible values of the label as $1, \ldots, C$.
  
  – Define matrix $Y$:
    
    $$ Y_{ij} = \begin{cases} 
    1 & \text{if } y_i = c, \\
    0 & \text{otherwise} 
    \end{cases} $$

  – This defines $C$ independent regression problems; solving them with least squares yields
    
    $$ \hat{Y}_0 = X_0 (X^T X)^{-1} XY. $$
Classification as regression

• Suppose we have a binary problem, \( y \in \{-1, 1\} \).

• Assuming the standard model \( y = f(x; w) + \nu \), and solving with least squares, we get \( \hat{w} \).

• This corresponds to squared loss as a measure of classification performance! Does this make sense?
Classification as regression

- Suppose we have a binary problem, $y \in \{-1, 1\}$.

- Assuming the standard model $y = f(x; w) + \nu$, and solving with least squares, we get $\hat{w}$.

- This corresponds to squared loss as a measure of classification performance! Does this make sense?

- How do we decide on the label based on $f(x; \hat{w})$?
Classification as regression: example

- A 1D example:
Classification as regression: example

- A 1D example:
Classification as regression: example

- A 1D example:

\[ w_0 + w^T x \]
Classification as regression: example

- A 1D example:

\[ \hat{y} = \begin{cases} +1 & \text{if } w_0 + w^T x > 0 \\ -1 & \text{if } w_0 + w^T x < 0 \end{cases} \]
Classification as regression

\[ f(x; \hat{w}) = w_0 + \hat{w}^T x \]

- Can’t just take \( \hat{y} = f(x; \hat{w}) \) since it won’t be a valid label.

- A reasonable decision rule:

  \[ \hat{y} = \text{sign} (w_0 + \hat{w}^T x) \]

- This specifies a linear classifier:
  - The linear decision boundary (hyperplane) given by the equation \( w_0 + \hat{w}^T x = 0 \) separates the space into two “half-spaces”.
Classification as regression

Seems to work well here but not so well here?
Geometry of projections

- \( w^T x = 0 \): a line passing through the origin and orthogonal to \( w \)

- \( w^T x + w_0 = 0 \) shifts the line along \( w \).
Geometry of projections

- $w^T x = 0$: a line passing through the origin and orthogonal to $w$
- $w^T x + w_0 = 0$ shifts the line along $w$. 

$$w_0 + w^T x = 0$$
Geometry of projections

- $w^T x = 0$: a line passing through the origin and orthogonal to $w$
- $w^T x + w_0 = 0$ shifts the line along $w$. 

\[ w_0 + w^T x = 0 \]
Geometry of projections

- $\mathbf{w}^T \mathbf{x} = 0$: a line passing through the origin and orthogonal to $\mathbf{w}$
- $\mathbf{w}^T \mathbf{x} + w_0 = 0$ shifts the line along $\mathbf{w}$. 

\[ x_1 \]

\[ x_2 \]

\[ \mathbf{w} \]

\[ \mathbf{x}_0 \]

\[ \mathbf{x}_0 \perp \]

\[ \mathbf{w}_0 + \mathbf{w}^T \mathbf{x}_0 = 0 \]

\[ \frac{\mathbf{w}_0 + \mathbf{w}^T \mathbf{x}_0}{\|\mathbf{w}\|} \]
Geometry of projections

- \( \mathbf{w}^T \mathbf{x} = 0 \): a line passing through the origin and orthogonal to \( \mathbf{w} \)
- \( \mathbf{w}^T \mathbf{x} + w_0 = 0 \) shifts the line along \( \mathbf{w} \).

- \( x' \) is the projection of \( \mathbf{x} \) on \( \mathbf{w} \).
Geometry of projections

- $\mathbf{w}^T \mathbf{x} = 0$: a line passing through the origin and orthogonal to $\mathbf{w}$
- $\mathbf{w}^T \mathbf{x} + w_0 = 0$ shifts the line along $\mathbf{w}$.

- $\mathbf{x}'$ is the projection of $\mathbf{x}$ on $\mathbf{w}$.

- Set up a new 1D coordinate system: $\mathbf{x} \rightarrow (w_0 + \mathbf{x}^T \mathbf{x})/\|\mathbf{w}\|$.
Distribution in 1D projection

- Consider a projection given by $\mathbf{w}^T \mathbf{x} = 0$ (i.e., $\mathbf{w}$ is the normal)

- Each training point $\mathbf{x}_i$ is projected to a scalar $z_i = \mathbf{w}^T \mathbf{x}$.

- We can study how well the projected values corresponding to different classes are separated
  - This is a function of $\mathbf{w}$; some projections may be better than others.
Linear discriminant and dimensionality reduction

The **discriminant function** \( f(x; w) = w_0 + w^T x \) reduces the dimension of examples from \( d \) to 1:
Projections and classification

What objective are we optimizing the 1D projection for?
1D projections of a Gaussian

- Let $p(x) = \mathcal{N}(x; \mu, \Sigma)$.

- For any $A$, $p(Ax) = \mathcal{N}(Ax; A\mu, A\Sigma A^T)$.

- To get a marginal of 1D projection on the direction defined by a unit vector $v$:
  - Make $R$ a rotation such that $R[1, 0, \ldots, 0]^T = v$
  - Compute $\Sigma_v = v^T \Sigma v$; that’s the variance of the marginal.
  - Let’s assume for now $\mu = 0$ (but think what happens if it’s not!)

- Matlab demo: margGausDemo.m
Objective: class separation

- We want to minimize “overlap” between projections of the two classes.
- One way to approach that: make the class projections a) compact, b) far apart.
Next time

Continue with linear discriminant analysis, and talk about optimal way to place the decision boundary.