What about prior

- Billionaire says: Wait, I know that the thumbtack is “close” to 50-50. What can you do for me now?
- **You say: I can learn it the Bayesian way…**

- Rather than estimating a single $\theta$, we obtain a distribution over possible values of $\theta$
Bayesian Learning

- Use Bayes rule:
  \[ P(\theta \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid \theta)P(\theta)}{P(\mathcal{D})} \]

- Or equivalently:
  \[ P(\theta \mid \mathcal{D}) \propto P(\mathcal{D} \mid \theta)P(\theta) \]

- MLE: \( \max_{\theta} P(\mathcal{D} \mid \theta) \)  
  \( \Rightarrow \theta^\text{MLE} \)

- Bayesian: \( P(\theta \mid \mathcal{D}) \)
  - When \( P(\theta) \) is uniform:
    \[ P(\theta \mid \mathcal{D}) \propto P(\mathcal{D} \mid \theta) \]

Bayesian Learning for Thumbtack

- Likelihood function is simply Binomial:
  \[ P(\mathcal{D} \mid \theta) = \theta^H (1 - \theta)^T \]

- What about prior?
  - Represent expert knowledge
  - Simple posterior form

- Conjugate priors:
  - Closed-form representation of posterior
  - For Binomial, conjugate prior is Beta distribution
Beta prior distribution – $P(\theta)$

$$P(\theta) = \frac{\theta^{\beta_H-1}(1-\theta)^{\beta_T-1}}{B(\beta_H, \beta_T)} \sim \text{Beta}(\beta_H, \beta_T)$$

Mean: $B(\beta_H + \beta_T - 2, 2)$
Mode: $\left(\theta_H - 1\right)/(\alpha_H + \beta_T - 2)$

- Likelihood function: $P(D | \theta) = \theta^{\alpha_H}(1-\theta)^{\alpha_T}$
- Posterior: $P(\theta | D) \propto P(D | \theta) P(\theta)$

Prior: Beta$(\beta_H, \beta_T)$
Data: $\alpha_H$ heads and $\alpha_T$ tails

Posterior distribution:

$$P(\theta | D) \sim \text{Beta}(\beta_H + \alpha_H, \beta_T + \alpha_T)$$

Bayesian update
Using Bayesian posterior

- Posterior distribution:
  \[ P(\theta \mid D) \sim Beta(\beta_H + \alpha_H, \beta_T + \alpha_T) \]

- Bayesian inference:
  - No longer single parameter:
    \[ E[f(\theta)] = \int_0^1 f(\theta) P(\theta \mid D) d\theta \]
  - Integral is often hard to compute

MAP: Maximum a posteriori approximation

- Posterior distribution:
  \[ P(\theta \mid D) \sim Beta(\beta_H + \alpha_H, \beta_T + \alpha_T) \]

- As more data is observed, Beta is more certain

- MAP: use most likely parameter:
  \[ \hat{\theta}_{MAP} = \arg \max_\theta P(\theta \mid D) \quad E[f(\theta)] \approx f(\hat{\theta}_{MAP}) \]
MAP for Beta distribution

\[ P(\theta \mid D) = \frac{\theta^{\beta_H + \alpha_H - 1} (1 - \theta)^{\beta_T + \alpha_T - 1}}{B(\beta_H + \alpha_H, \beta_T + \alpha_T)} \sim \text{Beta}(\beta_H + \alpha_H, \beta_T + \alpha_T) \]

- MAP: use most likely parameter:
  \[ \hat{\theta}_{\text{MAP}} = \arg \max \theta P(\theta \mid D) = \frac{\beta_H + \alpha_H - 1}{\beta_H + \alpha_H + \beta_T + \alpha_T - 2} \]

- Beta prior equivalent to extra thumbtack flips
- As \( N \to \frac{1}{2} \), prior is “forgotten” \( \alpha_H, \alpha_T \)
- But, for small sample size, prior is important!

---

Linear Regression

Machine Learning – CSE546
Carlos Guestrin
University of Washington
September 30, 2013
Prediction of continuous variables

- Billionaire sayz: Wait, that’s not what I meant!
- You sayz: Chill out, dude.
- He sayz: I want to predict a continuous variable for continuous inputs: I want to predict salaries from GPA.
- You sayz: I can regress that…

The regression problem

- Instances: \( \langle x_j, t_j \rangle \)
- Learn: Mapping from \( x \) to \( t(x) \)
- Hypothesis space:
  - Given, basis functions \( H = \{ h_1, \ldots, h_K \} \)
  - Find coeffs \( w = (w_1, \ldots, w_k) \)
  - \( t(x) = \sum_i w_i h_i(x) \)
  - Why is this called linear regression???
    - model is linear in the parameters

- Precisely, minimize the residual squared error:

\[
w^* = \arg \min_w \sum_{j=1}^p \left( t(x_j) - \sum_{i=1}^K w_i h_i(x_j) \right)^2
\]
The regression problem in matrix notation

\[ w^* = \arg \min_w \sum_j \left( t(x_j) - \sum_i w_i h_i(x_j) \right)^2 \]

\[ w^* = \arg \min_w (Hw - t)^T (Hw - t) \]

Minimizing the Residual

\[ w^* = \arg \min_w (Hw - t)^T (Hw - t) \]

\[ \nabla f(w) = 0 \implies \nabla H^T (Hw - t) = 0 \]

\[ H^T H w - H^T t = 0 \]

\[ \implies w^* = \left( H^T H \right)^{-1} H^T t \]

\[ \text{in scalar calculus,} \]

\[ \frac{\partial}{\partial w} (dw-t)(dw-t) = 2dw - t \]

\[ = 2\left( Hw - t \right) \]

\[ \text{in matrix calculus,} \]

\[ \text{let } \left[ (Hw - t)^T (Hw - t) \right] \]

\[ \text{is } 2H^T (Hw - t) \]
Regression solution = simple matrix operations

\[ w^* = \arg \min_w (Hw - t)^T (Hw - t) \]

residual error

solution: \[ w^* = (H^T H)^{-1} H^T t = A^{-1} b \]

where \[ A = H^T H = \begin{bmatrix} \vdots \\ \vdots \end{bmatrix} \]

\[ b = H^T t = \begin{bmatrix} \vdots \\ \vdots \end{bmatrix} \]

But, why?

- Billionaire (again) says: Why sum squared error???
- You say: Gaussians, Dr. Gateson, Gaussians…
- Model: prediction is linear function plus Gaussian noise
  \[ t(x) = \sum_i w_i h_i(x) + \varepsilon_x \]
  \( \varepsilon_x \sim N(0, \sigma^2) \) \( \varepsilon_x \) is independent
  \( t(x) \) is independent
  \( N(\varepsilon_x; h_i(x), \sigma^2) \)

- Learn \( w \) using MLE
  \[ P(t \mid x, w, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{[t-\sum_i w_i h_i(x)]^2}{2\sigma^2}} \]
Maximizing log-likelihood

Maximize:
\[
\ln P(D | w, \sigma) = \ln \left( \frac{1}{\sigma \sqrt{2\pi}} \right)^N \prod_{j=1}^N e^{-\frac{1}{2\sigma^2} \left( t(y_j) - \sum_{i} w_i h_i(x_j) \right)^2} \]

Go to recitation!! 😊
- Tuesday, 5:30pm in LOW 101

First homework will go out today
- Due on October 14
- Start early!!
Bias-Variance Tradeoff

Machine Learning – CSE546
Carlos Guestrin
University of Washington
September 30, 2013

Bias-Variance tradeoff – Intuition

- Model too “simple” → does not fit the data well
  - A biased solution

- Model too complex → small changes to the data, solution changes a lot
  - A high-variance solution
(Squared) Bias of learner

- Given dataset $D$ with $N$ samples, learn function $h_D(x)$
  - If you sample a different dataset $D'$ with $N$ samples, you will learn different $h_D'(x)$
  - **Expected hypothesis**: $E_D[h_D(x)] = \tilde{h}_N(x)$
    - $\tilde{h}_N(x)$ is what I expect to learn
  - **Bias**: difference between what you expect to learn and truth
    - Measures how well you expect to represent true solution
    - Decreases with more complex model
    - Bias$^2$ at one point $x$: $\left( f(x) - \tilde{h}_N(x) \right)^2$
    - Average Bias$^2$:
      \[
      E_x \left[ \left( f(x) - \tilde{h}_N(x) \right)^2 \right]
      \]

Variance of learner

- Given dataset $D$ with $N$ samples, learn function $h_D(x)$
  - If you sample a different dataset $D'$ with $N$ samples, you will learn different $h_D'(x)$
  - **Variance**: difference between what you expect to learn and what you learn from a particular dataset
    - Measures how sensitive learner is to specific dataset
    - Decreases with simpler model
    - Variance at one point $x$: $E_D \left[ (h_D(x) - \tilde{h}_N(x))^2 \right]$
    - Average variance:
      \[
      E_x E_D \left[ (h_D(x) - \tilde{h}_N(x))^2 \right]
      \]
Bias-Variance Tradeoff

- Choice of hypothesis class introduces learning bias
  - More complex class → less bias
  - More complex class → more variance

Bias-Variance Decomposition of Error

\[ h_N(x) = E_D[h_D(x)] \]

- Expected mean squared error: \( \text{MSE} = E_D \left[ E_x \left[ (t(x) - h_D(x))^2 \right] \right] \)
- To simplify derivation, drop \( x \):
  \[ E_D \left[ (t - h_D)^2 \right] \]
- Expanding the square:
  \[ E_D \left[ (t - \tilde{h}_N + \tilde{h}_N - h_D)^2 \right] \]
  \[ = E_D \left[ (t - \tilde{h}_N)^2 + (\tilde{h}_N - h_D)^2 \right] + 2E_D \left[ (t - \tilde{h}_N)(\tilde{h}_N - h_D) \right] \]
  \[ \text{bias} \quad \text{variance} \]

Play with this hint:
\[ \tilde{h}_N = E_D[h_0] \]
All others are constants.
Moral of the Story: Bias-Variance Tradeoff Key in ML

- Error can be decomposed:
  \[ \text{MSE} = E_D \left[ E_x \left[ (t(x) - h_D(x))^2 \right] \right] \]
  \[ = E_x \left[ (t(x) - \bar{h}_N(x))^2 \right] + E_D \left[ E_x \left[ (\bar{h}(x) - h_D(x))^2 \right] \right] \]

- Choice of hypothesis class introduces learning bias
  - More complex class → less bias
  - More complex class → more variance

What you need to know

- Regression
  - Basis function = features
  - Optimizing sum squared error
  - Relationship between regression and Gaussians
- Bias-variance trade-off
- Play with Applet