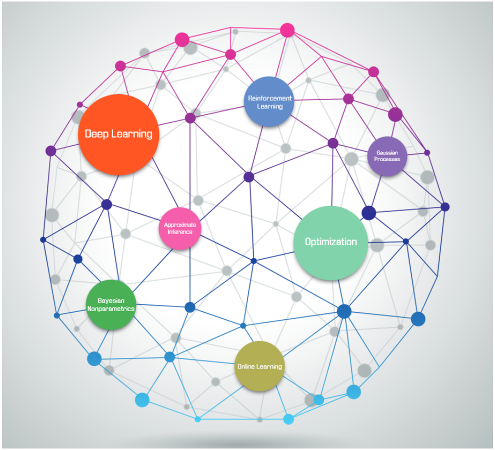
Bayesian Learning

Mark Schmidt

UBC Machine Learning Reading Group

January 2016

Current Hot Topics in Machine Learning



Bayesian learning includes:

- Gaussian processes.
- Approximate inference.
- Bayesian nonparametrics.

Why Bayesian Learning in the MLRG?

- Standard L2-regularized logistic regression steup:
  - Given finite dataset containing IID samples.
    - E.g., samples (x_i, y_i) with $x_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$.

Why Bayesian Learning in the MLRG?

- Standard L2-regularized logistic regression steup:
  - Given finite dataset containing IID samples.
    - E.g., samples (x_i, y_i) with $x_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$.
  - Predict label $\hat{y}$ of new example $\hat{x}$ using weights $\hat{w}$,

$$\hat{y} = \text{sgn}(\hat{w}^T \hat{x}).$$

Why Bayesian Learning in the MLRG?

- Standard L2-regularized logistic regression steup:
  - Given finite dataset containing IID samples.
    - E.g., samples (x_i, y_i) with $x_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$.
  - Predict label $\hat{y}$ of new example $\hat{x}$ using weights $\hat{w}$,

$$\hat{y} = \text{sgn}(\hat{w}^T \hat{x}).$$

  - Find 'best' $\hat{w}$ by minimizing loss function,

$$\hat{w} = \underset{w}{\text{argmin}} \sum_{i=1}^{n} \log(1 + \exp(-y_i w^T x_i)).$$

Why Bayesian Learning in the MLRG?

- Standard L2-regularized logistic regression steup:
  - Given finite dataset containing IID samples.
    - E.g., samples (x_i, y_i) with $x_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$.
  - Predict label $\hat{y}$ of new example $\hat{x}$ using weights $\hat{w}$,
$$\hat{y} = \text{sgn}(\hat{w}^T\hat{x}).$$
  - Find 'best' $\hat{w}$ by minimizing loss function,
$$\hat{w} = \underset{w}{\text{argmin}} \sum_{i=1}^{n} \log(1 + \exp(-y_i w^T x_i)).$$
  - Usually add regularization because it "prevents overfitting",
$$\hat{w} = \underset{w}{\text{argmin}} \sum_{i=1}^{n} \log(1 + \exp(-y_i w^T x_i)) + \frac{\lambda}{2}\|w\|^2.$$

Why Bayesian Learning in the MLRG?

- Standard L2-regularized logistic regression steup:
  - Given finite dataset containing IID samples.
    - E.g., samples (x_i, y_i) with $x_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$.
  - Predict label $\hat{y}$ of new example $\hat{x}$ using weights $\hat{w}$,

$$\hat{y} = \text{sgn}(\hat{w}^T \hat{x}).$$

  - Find 'best' $\hat{w}$ by minimizing loss function,

$$\hat{w} = \underset{w}{\text{argmin}} \sum_{i=1}^{n} \log(1 + \exp(-y_i w^T x_i)).$$

  - Usually add regularization because it "prevents overfitting",

$$\hat{w} = \underset{w}{\text{argmin}} \sum_{i=1}^{n} \log(1 + \exp(-y_i w^T x_i)) + \frac{\lambda}{2} \|w\|^2.$$

- Data was random, so weights $\hat{w}$ are random variables.
  - Finds $\hat{w}$ maximizing $p(\hat{w}|X, y)$, but predictions are sub-optimal.
    - Does not consider that $p(\hat{w}|X, y)$ may be tiny.

Why Bayesian Learning in the MLRG?

- Standard L2-regularized logistic regression steup:
  - Given finite dataset containing IID samples.
    - E.g., samples (x_i, y_i) with $x_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$.
  - Predict label $\hat{y}$ of new example $\hat{x}$ using weights $\hat{w}$,

$$\hat{y} = \text{sgn}(\hat{w}^T \hat{x}).$$

  - Find 'best' $\hat{w}$ by minimizing loss function,

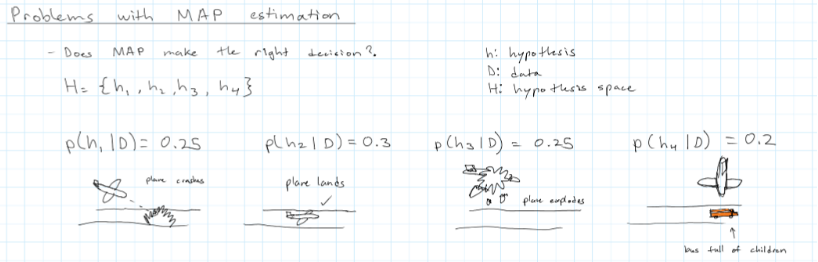
$$\hat{w} = \underset{w}{\text{argmin}} \sum_{i=1}^{n} \log(1 + \exp(-y_i w^T x_i)).$$

  - Usually add regularization because it "prevents overfitting",

$$\hat{w} = \underset{w}{\text{argmin}} \sum_{i=1}^{n} \log(1 + \exp(-y_i w^T x_i)) + \frac{\lambda}{2} \|w\|^2.$$

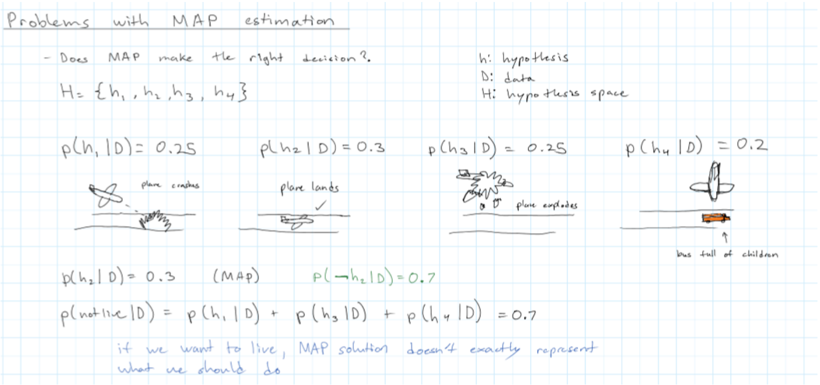
- Data was random, so weights $\hat{w}$ are random variables.
  - Finds $\hat{w}$ maximizing $p(\hat{w}|X, y)$, but predictions are sub-optimal.
    - Does not consider that $p(\hat{w}|X, y)$ may be tiny.
  - Bayesian approach: predictions based on rules of probability.

Why Bayesian Learning in the MLRG?



Optimization approach only considers h_2 so you should take plane.

Why Bayesian Learning in the MLRG?



Bayesian approach averages models: says you shouldn't take plane.

Bayesian decision theory: take into account cost of different errors.

Why Bayesian Learning in the MLRG?

- Motivation for studying Bayesian learning:
  1. Optimal decisions using rules of probability and error costs.

Why Bayesian Learning in the MLRG?

- Motivation for studying Bayesian learning:
  1. Optimal decisions using rules of probability and error costs.
  2. Gives estimates of variability/confidence.
     - E.g., this gene has a 70% chance of being relevant.

Why Bayesian Learning in the MLRG?

- Motivation for studying Bayesian learning:
  1. Optimal decisions using rules of probability and error costs.
  2. Gives estimates of variability/confidence.
     - E.g., this gene has a 70% chance of being relevant.
  3. Elegant approaches for model selection and model averaging.
     - E.g., optimize λ or optimize grouping of w elements.

Why Bayesian Learning in the MLRG?

- Motivation for studying Bayesian learning:
  1. Optimal decisions using rules of probability and error costs.
  2. Gives estimates of variability/confidence.
     - E.g., this gene has a 70% chance of being relevant.
  3. Elegant approaches for model selection and model averaging.
     - E.g., optimize λ or optimize grouping of w elements.
  4. Easy to relax IID assumption.
     - E.g., hierarchical Bayesian models for data from different sources.

Why Bayesian Learning in the MLRG?

- Motivation for studying Bayesian learning:
  1. Optimal decisions using rules of probability and error costs.
  2. Gives estimates of variability/confidence.
     - E.g., this gene has a 70% chance of being relevant.
  3. Elegant approaches for model selection and model averaging.
     - E.g., optimize λ or optimize grouping of w elements.
  4. Easy to relax IID assumption.
     - E.g., hierarchical Bayesian models for data from different sources.
  5. Bayesian optimization: fastest rates for some non-convex problems.

Why Bayesian Learning in the MLRG?

- Motivation for studying Bayesian learning:
  1. Optimal decisions using rules of probability and error costs.
  2. Gives estimates of variability/confidence.
     - E.g., this gene has a 70% chance of being relevant.
  3. Elegant approaches for model selection and model averaging.
     - E.g., optimize λ or optimize grouping of w elements.
  4. Easy to relax IID assumption.
     - E.g., hierarchical Bayesian models for data from different sources.
  5. Bayesian optimization: fastest rates for some non-convex problems.
  6. Allows models with unknown/infinite number of parameters.
     - E.g., number of clusters or number of states in hidden Markov model.

Why Bayesian Learning in the MLRG?

- Motivation for studying Bayesian learning:
  1. Optimal decisions using rules of probability and error costs.
  2. Gives estimates of variability/confidence.
     - E.g., this gene has a 70% chance of being relevant.
  3. Elegant approaches for model selection and model averaging.
     - E.g., optimize λ or optimize grouping of w elements.
  4. Easy to relax IID assumption.
     - E.g., hierarchical Bayesian models for data from different sources.
  5. Bayesian optimization: fastest rates for some non-convex problems.
  6. Allows models with unknown/infinite number of parameters.
     - E.g., number of clusters or number of states in hidden Markov model.
- Why isn't everyone using this?
  - Philosophical: Some people don't like "subjective" prior.
  - Computational: Typically leads to nasty integration problems.

Maximum Likelihood vs. Maximum a Posteriori (MAP)

- Maximum likelihood (least squares):

$$\hat{h} = \operatorname*{argmax}_{h \in \mathcal{H}} p(D|h) \qquad \text{(train)}$$

$$\hat{D} = \operatorname*{argmax}_{D} p(D|\hat{h}) \qquad \text{(predict)}$$

Could choose a very unlikely h that fits data well.

Maximum Likelihood vs. Maximum a Posteriori (MAP)

- Maximum likelihood (least squares):

$$\hat{h} = \underset{h \in \mathcal{H}}{\text{argmax}}\, p(D|h) \qquad \text{(train)}$$

$$\hat{D} = \underset{D}{\text{argmax}}\, p(D|\hat{h}) \qquad \text{(predict)}$$

Could choose a very unlikely h that fits data well.

- Maximum a posteriori (MAP) (regularized least squares):

$$\begin{aligned}\hat{h} &= \underset{h \in \mathcal{H}}{\text{argmax}}\, p(h|D) \\ &= \underset{h \in \mathcal{H}}{\text{argmax}}\, \frac{p(D|h)p(h)}{p(D)} && \text{(Bayes' rule)} \\ &= \underset{h \in \mathcal{H}}{\text{argmax}}\, p(D|h)p(h) && \text{(train)} \\ \hat{D} &= \underset{D}{\text{argmax}}\, p(D|\hat{h}) && \text{(predict)}\end{aligned}$$

Prior $p(h)$ penalizes unlikely hypotheses.

Digression: MAP vs. Regularized Optimization

- Consider MAP estimate conditioned on X for linear regression:
  - Data D is a set of n IID (x_i, y_i) samples stored in X and y.
  - Hypothesis h represented by a parameter vector w.
  - Hypothesis space $\mathcal{H}$ is $\mathbb{R}^d$.

Digression: MAP vs. Regularized Optimization

- Consider MAP estimate conditioned on X for linear regression:
  - Data D is a set of n IID (x_i, y_i) samples stored in X and y.
  - Hypothesis h represented by a parameter vector w.
  - Hypothesis space $\mathcal{H}$ is $\mathbb{R}^d$.

$$
\begin{aligned}
\hat{w} &= \operatorname*{argmax}_{w \in \mathbb{R}^d} p(w|X, y) && \text{(MAP def'n)} \\
&= \operatorname*{argmax}_{w \in \mathbb{R}^d} p(y|X, w)p(w) && \text{(Bayes', } w \perp X\text{)} \\
&= \operatorname*{argmax}_{w \in \mathbb{R}^d} \prod_{i=1}^{n} [p(y_i|x_i, w)]p(w) && \text{(IID assump)} \\
&= \operatorname*{argmax}_{w \in \mathbb{R}^d} \log \left(\prod_{i=1}^{n} [p(y_i|x_i, w)]p(w) \right) && \text{(log is monotonic)} \\
&= \operatorname*{argmax}_{w \in \mathbb{R}^d} \sum_{i=1}^{n} \log p(y_i|x_i, w) + \log p(w) && (\log(ab) = \log(a) + \log(b)) \\
&= \operatorname*{argmin}_{w \in \mathbb{R}^d} - \sum_{i=1}^{n} \log p(y_i|x_i, w) - \log p(w) && \text{(max = min\{neg\})}
\end{aligned}
$$

Digression: MAP vs. Regularized Optimization

- So MAP estimate can be written in the form

$$\hat{w} = \underset{w \in \mathbb{R}^d}{\text{argmin}} - \sum_{i=1}^{n} \log p(y_i | x_i, w) - \log p(w),$$

and by same argument maximum likelihood can be written

$$\hat{w} = \underset{w \in \mathbb{R}^d}{\text{argmin}} - \sum_{i=1}^{n} \log p(y_i | x_i, w),$$

Digression: MAP vs. Regularized Optimization

- So MAP estimate can be written in the form

$$\hat{w} = \underset{w \in \mathbb{R}^d}{\text{argmin}} - \sum_{i=1}^{n} \log p(y_i | x_i, w) - \log p(w),$$

  and by same argument maximum likelihood can be written

$$\hat{w} = \underset{w \in \mathbb{R}^d}{\text{argmin}} - \sum_{i=1}^{n} \log p(y_i | x_i, w),$$

- We obtain our standard models as special cases:
  - Least squares: $y_i \sim \mathcal{N}(w^T x_i, \sigma^2)$.
  - L2-regularized least squares: $y_i \sim \mathcal{N}(w^T x_i, \sigma^2), \quad w_j \sim \mathcal{N}(0, \frac{1}{\sqrt{\lambda}})$.
  - L2-regularized logistic regression:
    $y_i \sim \text{Sigm}(w^T x_i), \quad w_j \sim \mathcal{N}(0, \frac{1}{\sqrt{\lambda}})$.
  - L1-regularized logistic regression:
    $y_i \sim \text{Sigm}(w^T x_i), \quad w_j \sim \mathcal{L}(0, \frac{1}{\lambda})$.
  - And so on...

MAP vs. Bayes

- Maximum a posteriori (MAP) (regularized optimization):

$$\hat{h} = \operatorname*{argmax}_{h \in \mathcal{H}} p(h|D)$$

$$= \operatorname*{argmax}_{h \in \mathcal{H}} p(D|h)p(h) \qquad \text{(train)}$$

$$\hat{D} = \operatorname*{argmax}_{D} p(D|\hat{h}) \qquad \text{(predict)}$$

MAP vs. Bayes

- Maximum a posteriori (MAP) (regularized optimization):

$$\hat{h} = \underset{h \in \mathcal{H}}{\text{argmax}}\, p(h|D)$$
$$= \underset{h \in \mathcal{H}}{\text{argmax}}\, p(D|h)p(h) \qquad \text{(train)}$$
$$\hat{D} = \underset{D}{\text{argmax}}\, p(D|\hat{h}) \qquad \text{(predict)}$$

- Bayesian approach (Bayesian linear regression):
  - Predict by integrating over "hidden" parameters:

$$p(\hat{D}|D) = \int_{\mathcal{H}} p(\hat{D}, h|D)dh \qquad \text{(marginalization rule)}$$
$$= \int_{\mathcal{H}} p(\hat{D}|h, D)p(h|D)dh \qquad \text{(product rule)}$$
$$= \int_{\mathcal{H}} p(\hat{D}|h)p(h|D)dh \qquad (\text{assume } \hat{D} \perp D \mid h)$$

MAP vs. Bayes

- Maximum a posteriori (MAP) (regularized optimization):

$$\begin{aligned}\hat{h} &= \operatorname*{argmax}_{h \in \mathcal{H}} p(h|D) \\ &= \operatorname*{argmax}_{h \in \mathcal{H}} p(D|h)p(h) && \text{(train)} \\ \hat{D} &= \operatorname*{argmax}_{D} p(D|\hat{h}) && \text{(predict)}\end{aligned}$$

- Bayesian approach (Bayesian linear regression):
  - Predict by integrating over “hidden” parameters:

$$\begin{aligned}p(\hat{D}|D) &= \int_{\mathcal{H}} p(\hat{D}, h|D)dh && \text{(marginalization rule)} \\ &= \int_{\mathcal{H}} p(\hat{D}|h, D)p(h|D)dh && \text{(product rule)} \\ &= \int_{\mathcal{H}} p(\hat{D}|h)p(h|D)dh && (\text{assume } \hat{D} \perp D \mid h)\end{aligned}$$

  - Integrate over posterior distribution rather than optimize over it.
  - Note that $p(D|h)$ dominates $p(h|D)$ as datasize grows.

Coin Flipping Example: Model

3 ingredients for Bayesian analysis of coin flipping:

1. Use a Bernoulli likelihood for coin X landing 'heads',

$$p(X = `H' | \theta) = \theta, \quad p(X = `T' | \theta) = 1 - \theta,$$

Coin Flipping Example: Model

3 ingredients for Bayesian analysis of coin flipping:

1. Use a Bernoulli likelihood for coin X landing 'heads',

$$p(X = \text{`}H'|\theta) = \theta, \quad p(X = \text{`}T'|\theta) = 1 - \theta,$$

2. Our prior reflects our prior beliefs about θ, we'll assume:
   - The coin has a 50% chance of being fair ($\theta = 0.5$).
   - The coin has a 50% chance of being rigged ($\theta = 1$).

Coin Flipping Example: Model

3 ingredients for Bayesian analysis of coin flipping:

1. Use a Bernoulli likelihood for coin X landing 'heads',

$$p(X = \text{`}H'|\theta) = \theta, \quad p(X = \text{`}T'|\theta) = 1 - \theta,$$

2. Our prior reflects our prior beliefs about θ, we'll assume:
   - The coin has a 50% chance of being fair ($\theta = 0.5$).
   - The coin has a 50% chance of being rigged ($\theta = 1$).
3. Our data consists of three consecutive heads: 'HHH'.

Coin Flipping Example: Estimators

What is the probability that the next coin lands heads?

- Maximum likelihood estimate is $\hat{\theta} = 1$ since

$$1 = p(HHH|\theta = 1) > p(HHH|\theta = 0.5) = 1/8,$$

- MAP estimate is $\hat{\theta} = 1$ since

$$0.5 = p(HHH|\theta = 1)p(\theta = 1) > p(HHH|\theta = 0.5)p(\theta = 0.5) = 1/16,$$

- ML and MAP both the say probability is 1.
- But we believed that there was a 50% chance the coin is fair.

Coin Flipping Example: Posterior

What is the probability that the next coin lands heads?

- The posterior probability that $\theta = 1$ is

$$
\begin{aligned}
p(\theta = 1 | HHH) &= \frac{p(HHH|\theta = 1)p(\theta = 1)}{p(HHH)} \\
&= \frac{p(HHH|\theta = 1)p(\theta = 1)}{p(HHH|\theta = 0.5)p(\theta = 0.5) + p(HHH|\theta = 1)p(\theta = 1)} \\
&= \frac{(1)(0.5)}{(1/8)(0.5) + (1)(0.5)} = \frac{8}{9},
\end{aligned}
$$

and similarly we have $p(\theta = 0.5|HHH) = \frac{1}{9}$.

Coin Flipping Example: Posterior

What is the probability that the next coin lands heads?

- The posterior probability that $\theta = 1$ is

$$
\begin{aligned}
p(\theta = 1 | HHH) &= \frac{p(HHH|\theta = 1)p(\theta = 1)}{p(HHH)} \\
&= \frac{p(HHH|\theta = 1)p(\theta = 1)}{p(HHH|\theta = 0.5)p(\theta = 0.5) + p(HHH|\theta = 1)p(\theta = 1)} \\
&= \frac{(1)(0.5)}{(1/8)(0.5) + (1)(0.5)} = \frac{8}{9},
\end{aligned}
$$

and similarly we have $p(\theta = 0.5|HHH) = \frac{1}{9}$.

- Posterior predictive distribution is

$$
\begin{aligned}
&p(H|HHH) = p(H, \theta = 1|HHH) + p(H, \theta = 0.5|HHH) \\
&= p(H|\theta = 1, HHH)p(\theta = 1|HHH) + p(H|\theta = 0.5, HHH)p(\theta = 0.5|HH \\
&= p(H|\theta = 1)p(\theta = 1|HHH) + p(H|\theta = 0.5)p(\theta = 0.5|HHH) \\
&= (1)(8/9) + (0.5)(1/9) = 0.94.
\end{aligned}
$$

Coin Flipping Example: Discussion

Comments on coin flipping example:

- Bayesian prediction uses that HHH could come from fair coin.

Coin Flipping Example: Discussion

Comments on coin flipping example:

- Bayesian prediction uses that HHH could come from fair coin.
- As we see more heads, posterior converges to 1.
  - ML/MLE/Bayes usually agree as data size increases.
- If we ever see a tail, posterior of $\theta = 1$ becomes 0.

Coin Flipping Example: Discussion

Comments on coin flipping example:

- Bayesian prediction uses that HHH could come from fair coin.
- As we see more heads, posterior converges to 1.
  - ML/MLE/Bayes usually agree as data size increases.
- If we ever see a tail, posterior of $\theta = 1$ becomes 0.
- If the prior is correct, then Bayesian estimate is optimal:
  - Bayesian decision theory gives optimal action incorporating costs.

Coin Flipping Example: Discussion

Comments on coin flipping example:

- Bayesian prediction uses that HHH could come from fair coin.
- As we see more heads, posterior converges to 1.
  - ML/MLE/Bayes usually agree as data size increases.
- If we ever see a tail, posterior of $\theta = 1$ becomes 0.
- If the prior is correct, then Bayesian estimate is optimal:
  - Bayesian decision theory gives optimal action incorporating costs.
- If the prior is incorrect, Bayesian estimate may be worse.
  - This is where people get uncomfortable about “subjective” priors.
- But ML/MAP are also based on “subjective” assumptions.

Summary

- Summary of topics discussed this week:
  - Regularized optimization is usually equivalent to MAP estimation.
  - But MAP estimation is sub-optimal.
  - Bayesian methods give optimal estimators:
    - Integrate over posterior rather than maximize over the posterior.
  - But Bayesian methods require prior beliefs.

Summary

- Summary of topics discussed this week:
  - Regularized optimization is usually equivalent to MAP estimation.
  - But MAP estimation is sub-optimal.
  - Bayesian methods give optimal estimators:
    - Integrate over posterior rather than maximize over the posterior.
  - But Bayesian methods require prior beliefs.
- Topics for next week:
  - When can we compute the posterior predictive?
  - Are there “non-informative” priors?

Schedule

Jan 6	Baysics	Mark
Jan 13	Conjugate Priors, Non-Informative Priors	Nasim
Jan 20	Hierarchical Modeling and Bayesian Model Selection	Geoff
Jan 27	Gaussian Processes and Empirical Bayes	Issam
Feb 3	Basic Monte Carlo Methods	Ricky
Feb 10	MCMC	Jason
Feb 24	Bayesian Optimization	Hamed
Mar 2	Variational Bayes	Sharan
Mar 9	Stochastic Variational Inference	Reza
Mar 16	Non-Parametric Bayes 1	Mark
Mary 23	Non-Parametric Bayes 2	Reza
Mar 30	Expectation Propagation	Behrooz
Apr 6	Sequential Monte Carlo and Population MCMC	Julieta
Apr 13	Reversible-Jump MCMC	Rudy
Apr 20	Approximate Bayesian Computation	Alireza