CPSC 540: Machine Learning

Conditional Random Fields and Variational Inference

Mark Schmidt

University of British Columbia

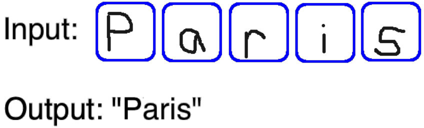
Winter 2016

Admin

- A5 posted, due April 12.
- Project:
  - Due date moved to April 29, description coming by April 12.

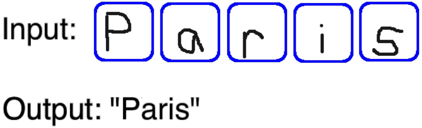
Structured Prediction with Undirected Graphical Models

- Recall the structured prediction problem:



Structured Prediction with Undirected Graphical Models

- Recall the structured prediction problem:

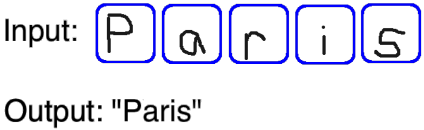


- We can view this as conditional density estimation,

$$p(Y|X) = \frac{\exp(-E(Y|X))}{Z},$$

Structured Prediction with Undirected Graphical Models

- Recall the structured prediction problem:



- We can view this as conditional density estimation,

$$p(Y|X) = \frac{\exp(-E(Y|X))}{Z},$$

where we've defined an energy function $E(Y|X)$:
  - Want low energy for correct labels.
  - Energy will depend on features $F(Y, X)$.
  - Usually energy is sum of parts, so we get a UGM

Structured Prediction with Undirected Graphical Models

- We might use an energy function with unary and pairwise terms,

$$E(Y|X) = -\sum_{j=1}^{d} \log \phi_j(y_j, X) - \sum_{(i,j)\in\mathcal{E}} \log \phi_{ij}(y_i, y_j, X),$$

Structured Prediction with Undirected Graphical Models

- We might use an energy function with unary and pairwise terms,

$$E(Y|X) = -\sum_{j=1}^{d} \log \phi_j(y_j, X) - \sum_{(i,j)\in\mathcal{E}} \log \phi_{ij}(y_i, y_j, X),$$

giving us a pairwise conditional UGM

$$p(Y|X) = \frac{\prod_{j=1}^{d} \phi_j(y_j, X) \prod_{ij} \phi_{ij}(y_i, y_j, X)}{Z}.$$

(we're treating X as fixed observations, not random variables)

Structured Prediction with Undirected Graphical Models

- We might use an energy function with unary and pairwise terms,

$$E(Y|X) = -\sum_{j=1}^{d} \log \phi_j(y_j, X) - \sum_{(i,j)\in\mathcal{E}} \log \phi_{ij}(y_i, y_j, X),$$

  giving us a pairwise conditional UGM

$$p(Y|X) = \frac{\prod_{j=1}^{d} \phi_j(y_j, X) \prod_{ij} \phi_{ij}(y_i, y_j, X)}{Z}.$$

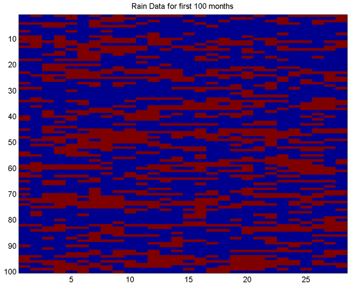
  (we're treating X as fixed observations, not random variables)
- Previously we focused on inference in UGMs:
  - We've discussed decoding, inference, and sampling.
- Today: learning the potential functions ϕ.
  - We'll start with the unconditional case (no X).

Example: Vancouver Rain Data

- Vancouver Rain data:
  - 1059 training examples x^i each containing 28 variables.
  - Variable x_j^i is whether or not it rained on day j in month i.
  - Data ranges from 1896-2004.

Example: Vancouver Rain Data

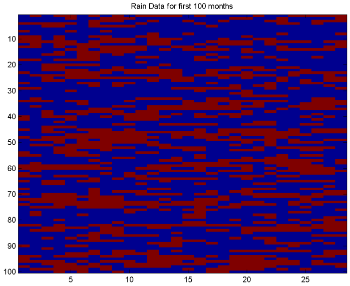
- Vancouver Rain data:
  - 1059 training examples x^i each containing 28 variables.
  - Variable x_j^i is whether or not it rained on day j in month i.
  - Data ranges from 1896-2004.
  - First 100 months (red means rain):

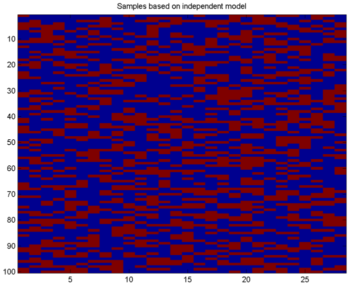


  - Sadly, $p(x_i = r) = 0.41$.

Example: Vancouver Rain Data

Real data vs. sampling day indepenedently with probability 0.41:

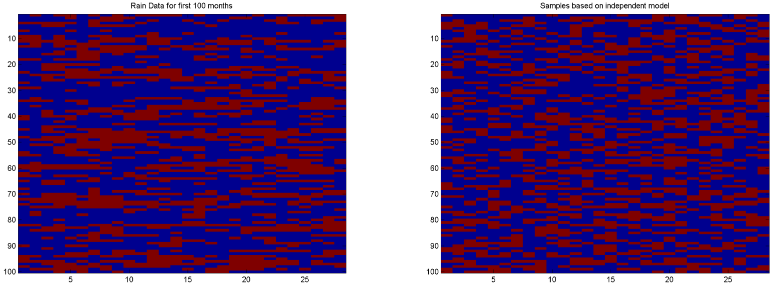




- Independent model misses correlations between days.

Example: Vancouver Rain Data

Real data vs. sampling day indepenedently with probability 0.41:



- Independent model misses correlations between days.
- We can do better with a UGM:
  - Assume we have a parameterization of our potentials.
  - Assume we use a chain-structured graph.
  - Output is the 'best' parameters (e.g., maximum likelihood).

Maximum Likelihood Formulation

- Let's fit the parameters using maximum likelihood of data:

(assuming the X^i are independent)

$$w = \underset{w}{\operatorname{argmax}} \prod_{i=1}^{n} p(X^i|w),$$

Maximum Likelihood Formulation

- Let's fit the parameters using maximum likelihood of data:

  (assuming the X^i are independent)

$$w = \underset{w}{\operatorname{argmax}} \prod_{i=1}^{n} p(X^i|w),$$

or equivalently minimize negative log-likelihood (NLL),

$$w = \underset{w}{\operatorname{argmin}} -\frac{1}{n}\sum_{i=1}^{n} \log(p(X^i|w)),$$

Maximum Likelihood Formulation

- Let's fit the parameters using maximum likelihood of data:

  (assuming the X^i are independent)

$$w = \underset{w}{\text{argmax}} \prod_{i=1}^{n} p(X^i|w),$$

or equivalently minimize negative log-likelihood (NLL),

$$w = \underset{w}{\text{argmin}} -\frac{1}{n}\sum_{i=1}^{n} \log(p(X^i|w)),$$

and you could/should also use a regularizer,

$$w = \underset{w}{\text{argmin}} -\frac{1}{n}\sum_{i=1}^{n} \log(p(X^i|w)) + \frac{\lambda}{2}\|w\|^2.$$

Log-Linear Parameterization of MRFs

- Naive parameterization:

$$\phi_i(x_i) = w_i, \quad \phi_{ij}(x_i, x_j) = w_{ij}.$$

  subject to $w \geq 0$.
- Not convex, and assumes potentials are all different.

Log-Linear Parameterization of MRFs

- Naive parameterization:

$$\phi_i(x_i) = w_i, \quad \phi_{ij}(x_i, x_j) = w_{ij}.$$

  subject to $w \geq 0$.
- Not convex, and assumes potentials are all different.
- We'll use a log-linear parameterization:

$$\phi_i(x_i) = \exp(w_{m(i,x_i)}), \quad \phi_{ij}(x_i, x_j) = \exp(w_{m(i,j,x_i,x_j)}).$$

  where m maps from parameters to potentials.

Log-Linear Parameterization of MRFs

- Naive parameterization:

$$\phi_i(x_i) = w_i, \quad \phi_{ij}(x_i, x_j) = w_{ij}.$$

  subject to $w \geq 0$.
- Not convex, and assumes potentials are all different.
- We'll use a log-linear parameterization:

$$\phi_i(x_i) = \exp(w_{m(i,x_i)}), \quad \phi_{ij}(x_i, x_j) = \exp(w_{m(i,j,x_i,x_j)}).$$

  where m maps from parameters to potentials.
- Parameter tieing can be done with choice of m:

Log-Linear Parameterization of MRFs

- Naive parameterization:

$$\phi_i(x_i) = w_i, \quad \phi_{ij}(x_i, x_j) = w_{ij}.$$

  subject to $w \geq 0$.
- Not convex, and assumes potentials are all different.
- We'll use a log-linear parameterization:

$$\phi_i(x_i) = \exp(w_{m(i,x_i)}), \quad \phi_{ij}(x_i, x_j) = \exp(w_{m(i,j,x_i,x_j)}).$$

  where m maps from parameters to potentials.
- Parameter tieing can be done with choice of m:
  - If $m(i, x_i) = x_i$ for all i, each day has same potentials.

(parameters are tied)

Log-Linear Parameterization of MRFs

- Naive parameterization:

$$\phi_i(x_i) = w_i, \quad \phi_{ij}(x_i, x_j) = w_{ij}.$$

  subject to $w \geq 0$.
- Not convex, and assumes potentials are all different.
- We'll use a log-linear parameterization:

$$\phi_i(x_i) = \exp(w_{m(i,x_i)}), \quad \phi_{ij}(x_i, x_j) = \exp(w_{m(i,j,x_i,x_j)}).$$

  where m maps from parameters to potentials.
- Parameter tieing can be done with choice of m:
  - If $m(i, x_i) = x_i$ for all i, each day has same potentials.

    (parameters are tied)
  - If $m(i, x_i) = x_i(n-1) + i$ for all i, each day has different potentials.

Log-Linear Parameterization of MRFs

- Naive parameterization:

$$\phi_i(x_i) = w_i, \quad \phi_{ij}(x_i, x_j) = w_{ij}.$$

  subject to $w \geq 0$.
- Not convex, and assumes potentials are all different.
- We'll use a log-linear parameterization:

$$\phi_i(x_i) = \exp(w_{m(i,x_i)}), \quad \phi_{ij}(x_i, x_j) = \exp(w_{m(i,j,x_i,x_j)}).$$

  where m maps from parameters to potentials.
- Parameter tieing can be done with choice of m:
  - If $m(i, x_i) = x_i$ for all i, each day has same potentials.
    (parameters are tied)
  - If $m(i, x_i) = x_i(n-1) + i$ for all i, each day has different potentials.
  - We could have groups: E.g., weekdays vs. weekends, or boundary.
  - We'll use the convention that $m(i, x_i) = 0$ means that $\phi_i(x_i) = 1$.

Log-Linear Parameterization of MRFs

- Naive parameterization:

$$\phi_i(x_i) = w_i, \quad \phi_{ij}(x_i, x_j) = w_{ij}.$$

  subject to $w \geq 0$.
- Not convex, and assumes potentials are all different.
- We'll use a log-linear parameterization:

$$\phi_i(x_i) = \exp(w_{m(i,x_i)}), \quad \phi_{ij}(x_i, x_j) = \exp(w_{m(i,j,x_i,x_j)}).$$

  where m maps from parameters to potentials.
- Parameter tieing can be done with choice of m:
  - If $m(i, x_i) = x_i$ for all i, each day has same potentials.

    (parameters are tied)
  - If $m(i, x_i) = x_i(n-1) + i$ for all i, each day has different potentials.
  - We could have groups: E.g., weekdays vs. weekends, or boundary.
  - We'll use the convention that $m(i, x_i) = 0$ means that $\phi_i(x_i) = 1$.
  - Similar logic holds for edge potentials.

Example: Ising Model of Rain Data

- E.g., we could parameterize our node potentials using

$$\log(\phi_i(x_i)) = \begin{cases} w_1 & \text{no rain} \\ 0 & \text{rain} \end{cases},$$

and one parameter is enough since scale of ϕ_i is arbitrary.

(though might want two parameters if using regularization)

Example: Ising Model of Rain Data

- E.g., we could parameterize our node potentials using

$$\log(\phi_i(x_i)) = \begin{cases} w_1 & \text{no rain} \\ 0 & \text{rain} \end{cases},$$

  and one parameter is enough since scale of ϕ_i is arbitrary.

  (though might want two parameters if using regularization)

- Ising parameterization of edge potentials,

$$\log(\phi_{ij}(x_i, x_j)) = \begin{cases} w_2 & x_i = x_j \\ 0 & x_i \neq x_j \end{cases}.$$

Example: Ising Model of Rain Data

- E.g., we could parameterize our node potentials using

$$\log(\phi_i(x_i)) = \begin{cases} w_1 & \text{no rain} \\ 0 & \text{rain} \end{cases},$$

  and one parameter is enough since scale of ϕ_i is arbitrary.

  (though might want two parameters if using regularization)

- Ising parameterization of edge potentials,

$$\log(\phi_{ij}(x_i, x_j)) = \begin{cases} w_2 & x_i = x_j \\ 0 & x_i \neq x_j \end{cases}.$$

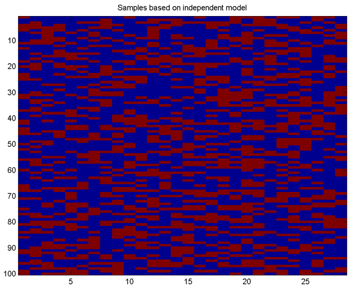
- Apply gradient descent to get maximum likelihood solution of

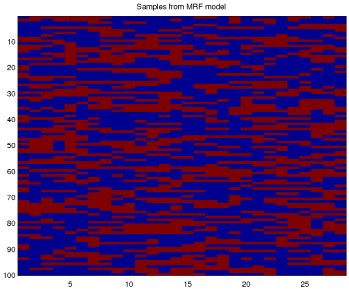
$$w = \begin{bmatrix} 0.16 \\ 0.85 \end{bmatrix}, \quad \phi_i = \begin{bmatrix} \exp(w_1) \\ \exp(0) \end{bmatrix} = \begin{bmatrix} 1.17 \\ 1 \end{bmatrix}, \quad \phi_{ij} = \begin{bmatrix} 2.34 & 1 \\ 1 & 2.34 \end{bmatrix},$$

  preference towards no rain, and adjacent days being the same.
- Average NLL of 16.8 vs. 19.0 for independent model.

Example: Ising Model of Rain Data

Independent model vs. Ising chain-UGM model:





Full Model of Rain Data

- We could alternately use fully expressive edge potentials

$$\log(\phi_{ij}(x_i, x_j)) = \begin{bmatrix} w_2 & w_3 \\ w_4 & w_5 \end{bmatrix},$$

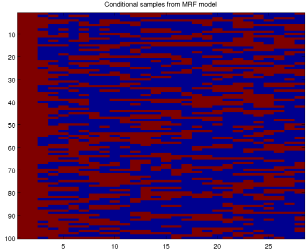
  but these don't improve the likelihood much.
  - Could also fix one of these at 0.
- We could also have special potentials for the boundaries.
  - Common in language models: treat start/end of setnence differently.

Full Model of Rain Data

- We could alternately use fully expressive edge potentials

$$\log(\phi_{ij}(x_i, x_j)) = \begin{bmatrix} w_2 & w_3 \\ w_4 & w_5 \end{bmatrix},$$

  but these don't improve the likelihood much.
  - Could also fix one of these at 0.
- We could also have special potentials for the boundaries.
  - Common in language models: treat start/end of setnence differently.
- Samples from model and conditional samples if rain on first day:



Log-Linear Parameterization of MRFs

- When we use a log-linear parameterization,

$$\phi_i(x_i) = \exp(w_{m(i,x_i)}), \quad \phi_{ij}(x_i, x_j) = \exp(w_{m(i,j,x_i,x_j)}),$$

  we exclude $\phi_i = 0$ but otherwise this is not restrictive.

Log-Linear Parameterization of MRFs

- When we use a log-linear parameterization,

$$\phi_i(x_i) = \exp(w_{m(i,x_i)}), \quad \phi_{ij}(x_i, x_j) = \exp(w_{m(i,j,x_i,x_j)}),$$

  we exclude $\phi_i = 0$ but otherwise this is not restrictive.
- Nice property: energy function $E(X)$ is linear,

$$\begin{aligned} E(X) &= \log\left(\prod_i \phi_i(x_i) \prod_{(i,j)\in E} \phi_{ij}(x_i, x_j)\right) \\ &= \log\left(\exp\left(\sum_i w_{m(i,x_i)} + \sum_{(i,j)\in E} w_{m(i,j,x_i,x_j)}\right)\right) \\ &= \sum_i w_{m(i,x_i)} + \sum_{(i,j)\in E} w_{m(i,j,x_i,x_j)}. \end{aligned}$$

Log-Linear Parameterization of MRFs

- When we use a log-linear parameterization,
$$\phi_i(x_i) = \exp(w_{m(i,x_i)}), \quad \phi_{ij}(x_i,x_j) = \exp(w_{m(i,j,x_i,x_j)}),$$
we exclude $\phi_i = 0$ but otherwise this is not restrictive.
- Nice property: energy function $E(X)$ is linear,
$$\begin{aligned} E(X) &= \log\left(\prod_i \phi_i(x_i) \prod_{(i,j)\in E} \phi_{ij}(x_i,x_j)\right) \\ &= \log\left(\exp\left(\sum_i w_{m(i,x_i)} + \sum_{(i,j)\in E} w_{m(i,j,x_i,x_j)}\right)\right) \\ &= \sum_i w_{m(i,x_i)} + \sum_{(i,j)\in E} w_{m(i,j,x_i,x_j)}. \end{aligned}$$
- To make notation simpler, consider this identity
$$w_{m(i,x_i)} = \sum_f w_f \mathcal{I}[m(i,x_i) = f],$$

Feature Vector Representation

- Use this identity to write any log-linear energy in a simple form

$$
\begin{aligned}
E(X) &= \sum_i w_{m(i,x_i)} + \sum_{(i,j)\in E} w_{m(i,j,x_i,x_j)} \\
&= \sum_i \sum_f w_f \mathcal{I}[m(i,x_i) = f] + \sum_{(i,j)\in E} \sum_f w_f \mathcal{I}[m(i,j,x_i,x_j) = f] \\
&= \sum_f w_f \left(\sum_i \mathcal{I}[m(i,x_i) = f] + \sum_{(i,j)\in E} \mathcal{I}[m(i,j,x_i,x_j) = f] \right) \\
&= w^T F(X)
\end{aligned}
$$

Feature Vector Representation

- Use this identity to write any log-linear energy in a simple form

$$
\begin{aligned}
E(X) &= \sum_i w_{m(i,x_i)} + \sum_{(i,j)\in E} w_{m(i,j,x_i,x_j)} \\
&= \sum_i \sum_f w_f \mathcal{I}[m(i,x_i) = f] + \sum_{(i,j)\in E} \sum_f w_f \mathcal{I}[m(i,j,x_i,x_j) = f] \\
&= \sum_f w_f \left(\sum_i \mathcal{I}[m(i,x_i) = f] + \sum_{(i,j)\in E} \mathcal{I}[m(i,j,x_i,x_j) = f] \right) \\
&= w^T F(X)
\end{aligned}
$$

- So $p(X) \propto \exp(E(X)) = \exp(w^T F(x))$ is in the exponential family.
- $F_f(X) \triangleq \sum_i \mathcal{I}[m(i,x_i) = f] + \sum_{(i,j)\in E} \mathcal{I}[m(i,j,x_i,x_j) = f]$ are sufficient statistics:
    - In Ising model $F_1(X)$ is number of times it rained in X and $F_2(X)$ is number adjacent days that have the same value.

MRF Training Objective Function

- With log-linear parameterization, NLL takes the form

$$
\begin{aligned}
f(w) &= -\frac{1}{n}\sum_{i=1}^{n} \log p(X^i|w) = -\frac{1}{n}\sum_{i=1}^{n} \log\left(\frac{\exp(w^T F(X^i))}{Z(w)}\right) \\
&= -\frac{1}{n}\sum_{i=1}^{n} w^T F(X^i) + \frac{1}{n}\sum_{i=1}^{n} \log Z(w) \\
&= -w^T F(D) + \log Z(w).
\end{aligned}
$$

where $F(D) = \frac{1}{n}\sum_i F(X^i)$ is sufficient statistics of data.

MRF Training Objective Function

- With log-linear parameterization, NLL takes the form

$$
\begin{aligned}
f(w) &= -\frac{1}{n}\sum_{i=1}^{n}\log p(X^i|w) = -\frac{1}{n}\sum_{i=1}^{n}\log\left(\frac{\exp(w^TF(X^i))}{Z(w)}\right)\\
&= -\frac{1}{n}\sum_{i=1}^{n} w^TF(X^i) + \frac{1}{n}\sum_{i=1}^{n}\log Z(w)\\
&= -w^TF(D) + \log Z(w).
\end{aligned}
$$

  where $F(D) = \frac{1}{n}\sum_i F(X^i)$ is sufficient statistics of data.
- Given sufficient statistics $F(D)$, can throw out data X^i.
  (only go through data once)
- Function $f(w)$ is convex.
- With $\|w\|^2$ regularizer, unique solution is guaranteed to exist.

Optimization with MRFs

- With log-linear parameterization, NLL takes the form

$$f(w) = -w^T F(D) + \log Z(w).$$

Optimization with MRFs

- With log-linear parameterization, NLL takes the form

$$f(w) = -w^T F(D) + \log Z(w).$$

- Gradient with respect to parameter f is given by

$$\begin{aligned} -\nabla_f f(w) &= F_f(D) - \sum_X \frac{\exp(w^T F(X))}{Z(w)} F_f(X) \\ &= F_f(D) - \sum_X p(X) F_f(X) \\ &= F_f(D) - \mathbb{E}_X[F_f(X)]. \end{aligned}$$

Optimization with MRFs

- With log-linear parameterization, NLL takes the form

$$f(w) = -w^T F(D) + \log Z(w).$$

- Gradient with respect to parameter f is given by

$$\begin{aligned} -\nabla_f f(w) &= F_f(D) - \sum_X \frac{\exp(w^T F(X))}{Z(w)} F_f(X) \\ &= F_f(D) - \sum_X p(X) F_f(X) \\ &= F_f(D) - \mathbb{E}_X[F_f(X)]. \end{aligned}$$

- Derivative of $\log(Z)$ is marginal of feature.
    - inference required for learning.

Optimization with MRFs

- With log-linear parameterization, NLL takes the form

$$f(w) = -w^T F(D) + \log Z(w).$$

- Gradient with respect to parameter f is given by

$$\begin{aligned}
-\nabla_f f(w) &= F_f(D) - \sum_X \frac{\exp(w^T F(X))}{Z(w)} F_f(X) \\
&= F_f(D) - \sum_X p(X) F_f(X) \\
&= F_f(D) - \mathbb{E}_X[F_f(X)].
\end{aligned}$$

- Derivative of $\log(Z)$ is marginal of feature.
    - inference required for learning.
- $\nabla_f f(w) = 0$ means sufficient statistics match in model and data.

Learning for Structured Prediction

3 types of classifiers discussed in CPSC 340/540:

Setting	Generative Model $p(Y, X)$	Discriminative Model $p(Y \mid X)$	Discriminant Function $Y = f(X)$
"Classic ML"	Naive Bayes, GDA	Logistic Regression	SVM

Learning for Structured Prediction

3 types of classifiers discussed in CPSC 340/540:

Setting	Generative Model $p(Y, X)$	Discriminative Model $p(Y \mid X)$	Discriminant Function $Y = f(X)$
"Classic ML"	Naive Bayes, GDA	Logistic Regression	SVM
Struct. Pred.	MRF	CRF	SSVM

Learning for Structured Prediction

3 types of classifiers discussed in CPSC 340/540:

Setting	Generative Model $p(Y, X)$	Discriminative Model $p(Y \mid X)$	Discriminant Function $Y = f(X)$
"Classic ML"	Naive Bayes, GDA	Logistic Regression	SVM
Struct. Pred.	MRF	CRF	SSVM

Generative models have lost popularity since modeling $p(X, Y)$ is harder than $p(Y \mid X)$.

Learning for Structured Prediction

3 types of classifiers discussed in CPSC 340/540:

Setting	Generative Model $p(Y, X)$	Discriminative Model $p(Y\|X)$	Discriminant Function $Y = f(X)$
"Classic ML"	Naive Bayes, GDA	Logistic Regression	SVM
Struct. Pred.	MRF	CRF	SSVM

Generative models have lost popularity since modeling $p(X, Y)$ is harder than $p(Y|X)$.
Has lead to rise in popularity of conditional models like CRFs:

- Directly model $p(Y|X)$ and just condition on X.
  - Extremely widely-used in natural language processing.
- I believe CRFs are second-most cited ML paper of 2000s:
  - 1. Topic models (non-parametric Bayes), 2. CRFs, 3. Deep learning.

Review of Discriminative Models for Classification

- Conditional random fields generalize logistic regression:

$$p(y = +1|x) = \frac{1}{1 + \exp(-yw^T x)} = \frac{\phi(+1)}{\phi(+1) + \phi(-1)}.$$

Review of Discriminative Models for Classification

- Conditional random fields generalize logistic regression:

$$p(y = +1|x) = \frac{1}{1 + \exp(-yw^T x)} = \frac{\phi(+1)}{\phi(+1) + \phi(-1)}.$$

$$\begin{aligned} p(y = -1|x) &= 1 - p(y = +1|x) = 1 - \frac{1}{1 + \exp(-yw^T x)} \\ &= \frac{\exp(-yw^T x)}{1 + \exp(-yw^T x)} = \frac{\phi(-1)}{\phi(+1) + \phi(-1)}. \end{aligned}$$

Review of Discriminative Models for Classification

- Conditional random fields generalize logistic regression:

$$p(y=+1|x) = \frac{1}{1+\exp(-yw^Tx)} = \frac{\phi(+1)}{\phi(+1)+\phi(-1)}.$$

$$\begin{aligned} p(y=-1|x) &= 1 - p(y=+1|x) = 1 - \frac{1}{1+\exp(-yw^Tx)} \\ &= \frac{\exp(-yw^Tx)}{1+\exp(-yw^Tx)} = \frac{\phi(-1)}{\phi(+1)+\phi(-1)}. \end{aligned}$$

- This is a conditional UGM with:

$$m(1,j,y=+1) = 0, \quad m(1,j,y=-1) = j.$$

Conditional Random Fields (CRFs)

- CRFs directly model $p(Y|X)$ for structured prediction
$$p(Y|X) = \frac{\exp(w^T F(Y,X))}{Z(w,X)},$$
where X is treated as fixed.
- Convex function and much simpler than generative approach:

Conditional Random Fields (CRFs)

- CRFs directly model $p(Y|X)$ for structured prediction
$$p(Y|X) = \frac{\exp(w^T F(Y,X))}{Z(w,X)},$$
where X is treated as fixed.
- Convex function and much simpler than generative approach:
  - No need to model features x for each possible object y.
- For pairwise UGMs, features have form $F(y_i, X)$ or $F(y_i, y_j, X)$.

Conditional Random Fields (CRFs)

- CRFs directly model $p(Y|X)$ for structured prediction

$$p(Y|X) = \frac{\exp(w^T F(Y,X))}{Z(w,X)},$$

where X is treated as fixed.

- Convex function and much simpler than generative approach:
  - No need to model features x for each possible object y.
- For pairwise UGMs, features have form $F(y_i, X)$ or $F(y_i, y_j, X)$.
- NLL and its gradient have similar form to MRFs

$$f(w) = -\frac{1}{n}\sum_{i=1}^{n} -w^T F(Y_i, X_i) + \log(Z(w, X_i)),$$

$$\nabla_f f(w) = -\frac{1}{n}\sum_{i=1}^{n} F(Y_i, X_i) + \mathbb{E}_{Y|X}[F_f(Y_i, X_i)],$$

but partition function and marginals for each example i.

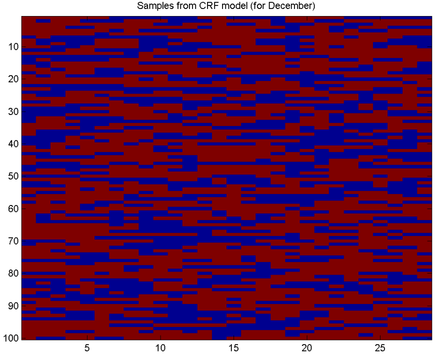
  - More expensive because don't have sufficient statistics.

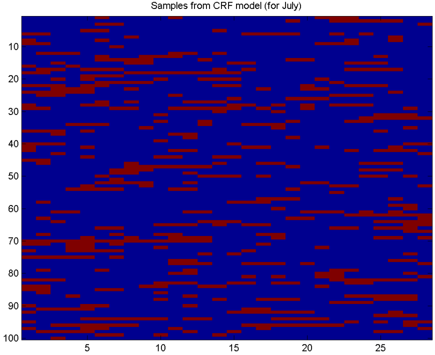
Rain Demo with Month Data

- Let's add a month variable to rain data:
  - Fit a CRF of p(rain | month).
  - Use 12 binary indicator features giving month.
  - NLL goes from 16.8 to 16.2.

Rain Demo with Month Data

- Let's add a month variable to rain data:
  - Fit a CRF of p(rain | month).
  - Use 12 binary indicator features giving month.
  - NLL goes from 16.8 to 16.2.
- Samples of rain data conditioned on December and July:





Approximate Learning

- Inference is a sub-routine of learning:
  - We can only learn when inference is tractable.

Approximate Learning

- Inference is a sub-routine of learning:
  - We can only learn when inference is tractable.
- Strategies when inference is not tractable:
  - Change the objective function:
    - Pseudo-likelihood (fast, convex, and crude):

$$\log p(Y|X) \approx \sum_i \log p(y_i|y_{-i}, X),$$

      transforms learning into logistic regression on each part.
    - SSVMs: generalization of SVMs that only requires decoding.

Approximate Learning

- Inference is a sub-routine of learning:
  - We can only learn when inference is tractable.
- Strategies when inference is not tractable:
  - Change the objective function:
    - Pseudo-likelihood (fast, convex, and crude):

      $$\log p(Y|X) \approx \sum_i \log p(y_i|y_{-i}, X),$$

      transforms learning into logistic regression on each part.
    - SSVMs: generalization of SVMs that only requires decoding.
  - Use approximate inference:
    - Monte Carlo methods.
    - Variational methods.

Outline

Variational Inference

- "Variational inference":
  - Formulate inference problem as constrained optimization.
  - Approximate the function or constraints to make it easy.

Variational Inference

- "Variational inference":
  - Formulate inference problem as constrained optimization.
  - Approximate the function or constraints to make it easy.
- Why not use MCMC?
  - MCMC works asymptotically, but may take forever.
  - Variational methods not consistent, but very fast.

(trade off accuracy vs. computation)

Exponential Families and Cumulant Function

- We will again consider log-linear models:

$$P(X) = \frac{\exp(w^T F(X))}{Z(w)},$$

but view them as exponential family distributions,

$$P(X) = \exp(w^T F(X) - A(w)),$$

where $A(w) = \log(Z(w))$.

Exponential Families and Cumulant Function

- We will again consider log-linear models:

$$P(X) = \frac{\exp(w^T F(X))}{Z(w)},$$

but view them as exponential family distributions,

$$P(X) = \exp(w^T F(X) - A(w)),$$

where $A(w) = \log(Z(w))$.

- Log-partition $A(w)$ is called the cumulant function,

$$\nabla A(w) = \mathbb{E}[F(X)], \quad \nabla^2 A(w) = \mathbb{V}[F(X)],$$

which implies convexity.

Convex Conjugate and Entropy

- The convex conjugate of a function A is given by

$$A^*(\mu) = \sup_{w \in \mathcal{W}} \{\mu^T w - A(w)\}.$$

Convex Conjugate and Entropy

- The convex conjugate of a function A is given by

$$A^*(\mu) = \sup_{w \in \mathcal{W}} \{\mu^T w - A(w)\}.$$

- E.g., in A3 we did this for logistic regression:

$$A(w) = \log(1 + \exp(w)),$$

Convex Conjugate and Entropy

- The convex conjugate of a function A is given by

$$A^*(\mu) = \sup_{w \in \mathcal{W}} \{\mu^T w - A(w)\}.$$

- E.g., in A3 we did this for logistic regression:

$$A(w) = \log(1 + \exp(w)),$$

implies that $A^*(\mu)$ satisfies $w = \log(\mu)/\log(1-\mu)$.

Convex Conjugate and Entropy

- The convex conjugate of a function A is given by

$$A^*(\mu) = \sup_{w \in \mathcal{W}} \{\mu^T w - A(w)\}.$$

- E.g., in A3 we did this for logistic regression:

$$A(w) = \log(1 + \exp(w)),$$

implies that $A^*(\mu)$ satisfies $w = \log(\mu)/\log(1-\mu)$.

  - When $0 < \mu < 1$ we have

$$\begin{aligned} A^*(\mu) &= \mu \log(\mu) + (1-\mu)\log(1-\mu) \\ &= -H(p_\mu), \end{aligned}$$

    negative entropy of binary distribution with mean μ.
  - If μ does not satisfy boundary constraint, $\sup$ is ∞.

Convex Conjugate and Entropy

- More generally, if $A(w) = \log(Z(w))$ then
$$A^*(\mu) = -H(p_\mu),$$
subject to boundary constraints on μ and constraint:
$$\mu = \nabla A(w) = \mathbb{E}[F(X)].$$
- Convex set satisfying these is called marginal polytope $\mathcal{M}$.

Convex Conjugate and Entropy

- More generally, if $A(w) = \log(Z(w))$ then

$$A^*(\mu) = -H(p_\mu),$$

subject to boundary constraints on μ and constraint:

$$\mu = \nabla A(w) = \mathbb{E}[F(X)].$$

- Convex set satisfying these is called marginal polytope $\mathcal{M}$.
- If A is convex (and LSC), $A^{**} = A$. So we have

$$A(w) = \sup_{\mu \in \mathcal{U}} \{w^T \mu - A^*(\mu)\}.$$

Convex Conjugate and Entropy

- More generally, if $A(w) = \log(Z(w))$ then

$$A^*(\mu) = -H(p_\mu),$$

subject to boundary constraints on μ and constraint:

$$\mu = \nabla A(w) = \mathbb{E}[F(X)].$$

- Convex set satisfying these is called marginal polytope $\mathcal{M}$.
- If A is convex (and LSC), $A^{**} = A$. So we have

$$A(w) = \sup_{\mu \in \mathcal{U}} \{w^T\mu - A^*(\mu)\}.$$

and when $A(w) = \log(Z(w))$ we have

$$\log(Z(w)) = \sup_{\mu \in \mathcal{M}} \{w^T\mu + H(p_\mu)\}.$$

- We've written inference as a convex optimization problem.

Bonus slide: Maximum Likelihood and Maximum Entropy

- The maximum likelihood parameters w satisfy:

$$
\begin{aligned}
&\min_{w \in \mathbb{R}^d} -w^T F(D) + \log(Z(w)) \\
&= \min_{w \in \mathbb{R}^d} -w^T F(D) + \sup_{\mu \in \mathcal{M}} \{w^T \mu + H(p_\mu)\} && \text{(convex conjugate)} \\
&= \min_{w \in \mathbb{R}^d} \sup_{\mu \in \mathcal{M}} \{-w^T F(D) + w^T \mu + H(p_\mu)\} \\
&= \sup_{\mu \in \mathcal{M}} \{ \min_{w \in \mathbb{R}^d} -w^T F(D) + w^T \mu + H(p_\mu)\} && \text{(convex/concave)}
\end{aligned}
$$

Bonus slide: Maximum Likelihood and Maximum Entropy

- The maximum likelihood parameters w satisfy:

$$
\begin{aligned}
&\min_{w\in\mathbb{R}^d} -w^T F(D) + \log(Z(w)) \\
&= \min_{w\in\mathbb{R}^d} -w^T F(D) + \sup_{\mu\in\mathcal{M}} \{w^T\mu + H(p_\mu)\} && \text{(convex conjugate)} \\
&= \min_{w\in\mathbb{R}^d} \sup_{\mu\in\mathcal{M}} \{-w^T F(D) + w^T\mu + H(p_\mu)\} \\
&= \sup_{\mu\in\mathcal{M}} \{ \min_{w\in\mathbb{R}^d} -w^T F(D) + w^T\mu + H(p_\mu)\} && \text{(convex/concave)}
\end{aligned}
$$

which is $-\infty$ unless $F(D) = \mu$ (e.g., maximum likelihood w), so we have

Bonus slide: Maximum Likelihood and Maximum Entropy

- The maximum likelihood parameters w satisfy:

$$\begin{aligned}
&\min_{w\in\mathbb{R}^d} -w^T F(D) + \log(Z(w)) \\
&= \min_{w\in\mathbb{R}^d} -w^T F(D) + \sup_{\mu\in\mathcal{M}} \{w^T\mu + H(p_\mu)\} && \text{(convex conjugate)} \\
&= \min_{w\in\mathbb{R}^d} \sup_{\mu\in\mathcal{M}} \{-w^T F(D) + w^T\mu + H(p_\mu)\} \\
&= \sup_{\mu\in\mathcal{M}} \{ \min_{w\in\mathbb{R}^d} -w^T F(D) + w^T\mu + H(p_\mu)\} && \text{(convex/concave)}
\end{aligned}$$

which is $-\infty$ unless $F(D) = \mu$ (e.g., maximum likelihood w), so we have

$$\begin{aligned}
&\min_{w\in\mathbb{R}^d} -w^T F(D) + \log(Z(w)) \\
&= \max_{\mu\in\mathcal{M}} H(p_\mu),
\end{aligned}$$

subject to $F(D) = \mu$.

Bonus slide: Maximum Likelihood and Maximum Entropy

- The maximum likelihood parameters w satisfy:

$$\begin{aligned}
&\min_{w\in\mathbb{R}^d} -w^T F(D) + \log(Z(w)) \\
&= \min_{w\in\mathbb{R}^d} -w^T F(D) + \sup_{\mu\in\mathcal{M}} \{w^T\mu + H(p_\mu)\} && \text{(convex conjugate)} \\
&= \min_{w\in\mathbb{R}^d} \sup_{\mu\in\mathcal{M}} \{-w^T F(D) + w^T\mu + H(p_\mu)\} \\
&= \sup_{\mu\in\mathcal{M}} \{\min_{w\in\mathbb{R}^d} -w^T F(D) + w^T\mu + H(p_\mu)\} && \text{(convex/concave)}
\end{aligned}$$

which is $-\infty$ unless $F(D) = \mu$ (e.g., maximum likelihood w), so we have

$$\begin{aligned}
&\min_{w\in\mathbb{R}^d} -w^T F(D) + \log(Z(w)) \\
&= \max_{\mu\in\mathcal{M}} H(p_\mu),
\end{aligned}$$

subject to $F(D) = \mu$.

- Maximum likelihood $\Rightarrow$ maximum entropy $+$ moment constraints.

Difficulty of Variational Formulation

- We wrote inference as a convex optimization:

$$\log(Z)) = \sup_{\mu \in \mathcal{M}} \{w^T \mu + H(p_\mu)\},$$

Difficulty of Variational Formulation

- We wrote inference as a convex optimization:

$$\log(Z)) = \sup_{\mu \in \mathcal{M}} \{w^T \mu + H(p_\mu)\},$$

- Did this make anything easier?

Difficulty of Variational Formulation

- We wrote inference as a convex optimization:

$$\log(Z)) = \sup_{\mu \in \mathcal{M}} \{w^T \mu + H(p_\mu)\},$$

- Did this make anything easier?
    - Computing entropy $H(p_\mu)$ seems as hard as inference.
    - Characterizing marginal polytope $\mathcal{M}$ becomes hard with loops.

Difficulty of Variational Formulation

- We wrote inference as a convex optimization:

$$\log(Z)) = \sup_{\mu \in \mathcal{M}} \{w^T \mu + H(p_\mu)\},$$

- Did this make anything easier?
  - Computing entropy $H(p_\mu)$ seems as hard as inference.
  - Characterizing marginal polytope $\mathcal{M}$ becomes hard with loops.
- Practical variational methods:
  - Work with approximation to marginal polytope $\mathcal{M}$.
  - Work with approximation/bound on entropy A^*.

Difficulty of Variational Formulation

- We wrote inference as a convex optimization:

$$\log(Z)) = \sup_{\mu \in \mathcal{M}} \{w^T \mu + H(p_\mu)\},$$

- Did this make anything easier?
  - Computing entropy $H(p_\mu)$ seems as hard as inference.
  - Characterizing marginal polytope $\mathcal{M}$ becomes hard with loops.
- Practical variational methods:
  - Work with approximation to marginal polytope $\mathcal{M}$.
  - Work with approximation/bound on entropy A^*.
- Notatation trick: we put everything "inside" w to discuss general log-potentials.

Mean Field Approximation

- Mean field approximation assumes

  $$\mu_{ij,st} = \mu_{i,s}\mu_{j,t},$$

  for all edges, which means

  $$p(x_i = s, x_j = t) = p(x_i = s)p(x_j = t),$$

  and that variables are independent.

Mean Field Approximation

- Mean field approximation assumes

$$\mu_{ij,st} = \mu_{i,s}\mu_{j,t},$$

  for all edges, which means

$$p(x_i = s, x_j = t) = p(x_i = s)p(x_j = t),$$

  and that variables are independent.
- Entropy is simple under mean field approximation:

$$\sum_X p(X)\log p(X) = \sum_i \sum_{x_i} p(x_i)\log p(x_i).$$

Mean Field Approximation

- Mean field approximation assumes

$$\mu_{ij,st} = \mu_{i,s}\mu_{j,t},$$

for all edges, which means

$$p(x_i = s, x_j = t) = p(x_i = s)p(x_j = t),$$

and that variables are independent.

- Entropy is simple under mean field approximation:

$$\sum_X p(X) \log p(X) = \sum_i \sum_{x_i} p(x_i) \log p(x_i).$$

- Marginal polytope is also simple:

$$\mathcal{M}_F = \{\mu \mid \mu_{i,s} \geq 0,\ \sum_s \mu_{i,s} = 1,\ \mu_{ij,st} = \mu_{i,s}\mu_{j,t}\}.$$

Bonus slide: Entropy of Mean Field Approximation

- Entropy form is from distributive law and probabilities sum to 1:

$$
\begin{aligned}
\sum_X p(X)\log p(X) &= \sum_X p(X)\log(\prod_i p(x_i)) \\
&= \sum_X p(X)\sum_i \log(p(x_i)) \\
&= \sum_i \sum_X p(X)\log p(x_i) \\
&= \sum_i \sum_X \prod_j p(x_j)\log p(x_i) \\
&= \sum_i \sum_X p(x_i)\log p(x_i)\prod_{j\neq i} p(x_j) \\
&= \sum_i \sum_{x_i} p(x_i)\log p(x_i) \sum_{x_j \,|\, j\neq i}\prod_{j\neq i} p(x_j) \\
&= \sum_i \sum_{x_i} p(x_i)\log p(x_i).
\end{aligned}
$$

Mean Field as Non-Convex Lower Bound

- Since $\mathcal{M}_F \subseteq \mathcal{M}$, yields a lower bound on $\log(Z)$:

$$\sup_{\mu \in \mathcal{M}_F} \{w^T \mu + H(p_\mu)\} \leq \sup_{\mu \in \mathcal{M}} \{w^T \mu + H(p_\mu)\} = \log(Z).$$

Mean Field as Non-Convex Lower Bound

- Since $\mathcal{M}_F \subseteq \mathcal{M}$, yields a lower bound on $\log(Z)$:

$$\sup_{\mu \in \mathcal{M}_F} \{w^T\mu + H(p_\mu)\} \leq \sup_{\mu \in \mathcal{M}} \{w^T\mu + H(p_\mu)\} = \log(Z).$$

- Since $\mathcal{M}_F \subseteq \mathcal{M}$, it is an inner approximation:

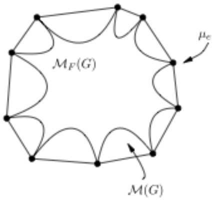


Fig. 5.3 Cartoon illustration of the set $\mathcal{M}_F(G)$ of mean parameters that arise from tractable distributions is a nonconvex inner bound on $\mathcal{M}(G)$. Illustrated here is the case of discrete random variables where $\mathcal{M}(G)$ is a polytope. The circles correspond to mean parameters that arise from delta distributions, and belong to both $\mathcal{M}(G)$ and $\mathcal{M}_F(G)$.

- Constraints $\mu_{ij,st} = \mu_{i,s}\mu_{j,t}$ make it non-convex.
- Mean field algorithm is coordinate descent on $w^T\mu + H(p_\mu)$ over $\mathcal{M}_F$.

Discussion of Mean Field and Structured MF

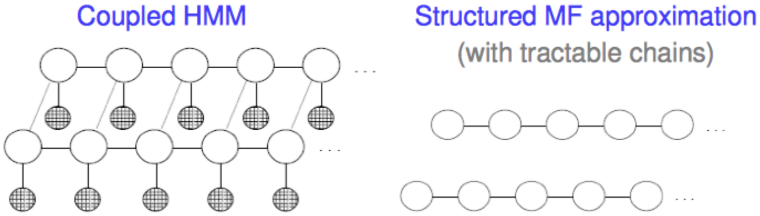
- Mean field is weird:
  - Non-convex approximation to a convex problem.
  - For learning, we want upper bounds on $\log(Z)$.

Discussion of Mean Field and Structured MF

- Mean field is weird:
  - Non-convex approximation to a convex problem.
  - For learning, we want upper bounds on $\log(Z)$.
- Structured mean field:
  - Cost of computing entropy is similar to cost of inference.

Discussion of Mean Field and Structured MF

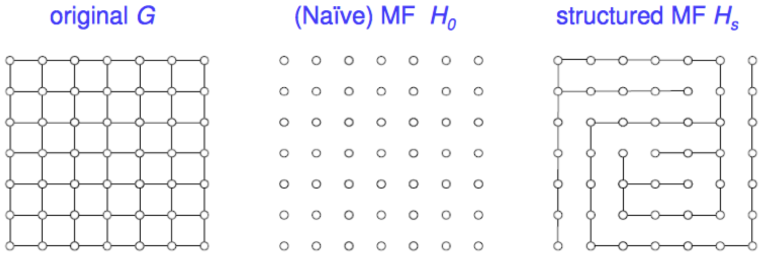
- Mean field is weird:
  - Non-convex approximation to a convex problem.
  - For learning, we want upper bounds on $\log(Z)$.
- Structured mean field:
  - Cost of computing entropy is similar to cost of inference.
  - Use a subgraph where we can perform exact inference.



http://courses.cms.caltech.edu/cs155/slides/cs155-14-variational.pdf

Structured Mean Field with Tree

More edges means better approximation of $\mathcal{M}$ and $H(p_\mu)$:



`http://courses.cms.caltech.edu/cs155/slides/cs155-14-variational.pdf`

Summary

- Log-linear parameterization can be used to learn UGMs:
  - Maximum likelihood is convex, but requires normalizing constant Z.
- Conditional random fields are UGMs that treat X as fixed and model $p(Y|X)$.
  - Log-linear parameterization again leads to convexity.
- Variational inference methods formulate counting/integrals as continuous optimization.
  - For UGMs, this is done via the convex conjugate.
  - Mean-field is one of the most common methods.

Next time: combining graphical models and deep learning.