SSVMs and BCFW

Mark Schmidt

University of British Columbia

August, 2015

Context: Structured Prediction with Log-Linear Models

- We used structured prediction to motivate studying UGMs:

Input: 

Output: "Paris"

Context: Structured Prediction with Log-Linear Models

- We used structured prediction to motivate studying UGMs:

Input: 

Output: "Paris"

- We talked about log-linear energy functions $E(Y|X)$:

$$E(Y|X) = w^T F(Y, X),$$

which makes maximum likelihood a convex problem.

Context: Structured Prediction with Log-Linear Models

- We used structured prediction to motivate studying UGMs:

Input: 

Output: "Paris"

- We talked about log-linear energy functions $E(Y|X)$:

$$E(Y|X) = w^T F(Y, X),$$

which makes maximum likelihood a convex problem.

- Today we consider an alternate learning principle:

Setting	Generative Model	Discriminative Model	Discriminant Function
"Classic ML"	Naive Bayes, GDA	Logistic Regression	SVM
Struct. Pred.	MRF	CRF	SSVM

SVMs and Structured SVMs

Insert first half of presentation from 2009...

SSVM Discussion

Beyond learning principle, key differences between CRFs/ SSVMs:

- SSVMs require decoding, not inference, for learning:
    - Exact SSVMs in cases like graph cuts, matchings, rankings, etc.

SSVM Discussion

Beyond learning principle, key differences between CRFs/ SSVMs:

- SSVMs require decoding, not inference, for learning:
  - Exact SSVMs in cases like graph cuts, matchings, rankings, etc.
- SSVMs have loss function for complicated accuracy measures:
  - But loss needs to decompose over parts for tractability.
  - Could also formulate 'loss-augmented' CRFs.

Fitting Structured SVMs

Overview of progress on training SSVMs:

- Cutting plane and bundle methods (e.g., svmStruct software):
  - Require $O(1/\epsilon)$ iterations.

Fitting Structured SVMs

Overview of progress on training SSVMs:

- Cutting plane and bundle methods (e.g., svmStruct software):
  - Require $O(1/\epsilon)$ iterations.
  - Each iteration requires decoding on every training example.

Fitting Structured SVMs

Overview of progress on training SSVMs:

- Cutting plane and bundle methods (e.g., svmStruct software):
  - Require $O(1/\epsilon)$ iterations.
  - Each iteration requires decoding on every training example.
- Stochastic sub-gradient methods:
  - Each iteration requires decoding on a single training example.
  - Still requires $O(1/\epsilon)$ iterations.

Fitting Structured SVMs

Overview of progress on training SSVMs:

- Cutting plane and bundle methods (e.g., svmStruct software):
  - Require $O(1/\epsilon)$ iterations.
  - Each iteration requires decoding on every training example.
- Stochastic sub-gradient methods:
  - Each iteration requires decoding on a single training example.
  - Still requires $O(1/\epsilon)$ iterations.
  - Need to choose step size.

Fitting Structured SVMs

Overview of progress on training SSVMs:

- Cutting plane and bundle methods (e.g., svmStruct software):
  - Require $O(1/\epsilon)$ iterations.
  - Each iteration requires decoding on every training example.
- Stochastic sub-gradient methods:
  - Each iteration requires decoding on a single training example.
  - Still requires $O(1/\epsilon)$ iterations.
  - Need to choose step size.
- Dual Online exponentiated gradient (OEG):
  - Allows line-search for step size and has $O(1/\epsilon)$ rate.

Fitting Structured SVMs

Overview of progress on training SSVMs:

- Cutting plane and bundle methods (e.g., svmStruct software):
  - Require $O(1/\epsilon)$ iterations.
  - Each iteration requires decoding on every training example.
- Stochastic sub-gradient methods:
  - Each iteration requires decoding on a single training example.
  - Still requires $O(1/\epsilon)$ iterations.
  - Need to choose step size.
- Dual Online exponentiated gradient (OEG):
  - Allows line-search for step size and has $O(1/\epsilon)$ rate.
  - Each iteration requires inference on a single training example.

Fitting Structured SVMs

Overview of progress on training SSVMs:

- Cutting plane and bundle methods (e.g., svmStruct software):
  - Require $O(1/\epsilon)$ iterations.
  - Each iteration requires decoding on every training example.
- Stochastic sub-gradient methods:
  - Each iteration requires decoding on a single training example.
  - Still requires $O(1/\epsilon)$ iterations.
  - Need to choose step size.
- Dual Online exponentiated gradient (OEG):
  - Allows line-search for step size and has $O(1/\epsilon)$ rate.
  - Each iteration requires inference on a single training example.
- Dual block-coordinate Frank-Wolfe (BCFW):
  - Each iteration requires decoding on a single training example.
  - Requires $O(1/\epsilon)$ iterations.
  - Closed-form optimal step size.

Fitting Structured SVMs

Overview of progress on training SSVMs:

- Cutting plane and bundle methods (e.g., svmStruct software):
  - Require $O(1/\epsilon)$ iterations.
  - Each iteration requires decoding on every training example.
- Stochastic sub-gradient methods:
  - Each iteration requires decoding on a single training example.
  - Still requires $O(1/\epsilon)$ iterations.
  - Need to choose step size.
- Dual Online exponentiated gradient (OEG):
  - Allows line-search for step size and has $O(1/\epsilon)$ rate.
  - Each iteration requires inference on a single training example.
- Dual block-coordinate Frank-Wolfe (BCFW):
  - Each iteration requires decoding on a single training example.
  - Requires $O(1/\epsilon)$ iterations.
  - Closed-form optimal step size.
  - Theory allows approximate decoding.

Block Coordinate Frank Wolfe

Key ideas behind BCFW for SSVMs:

- Dual problem has as the form

$$\min_{\alpha_i \in \mathcal{M}_i} F(\alpha) = f(A\alpha) - \sum_i f_i(\alpha_i).$$

  where f is smooth.

Block Coordinate Frank Wolfe

Key ideas behind BCFW for SSVMs:

- Dual problem has as the form

$$\min_{\alpha_i \in \mathcal{M}_i} F(\alpha) = f(A\alpha) - \sum_i f_i(\alpha_i).$$

  where f is smooth.

- Problem structure where we can use block coordinate descent:
  - Normal coordinate updates intractable because $\alpha_i \in |\mathcal{Y}|$.
  - But Frank-Wolfe block-coordinate update is equivalent to decoding

$$s = \operatorname*{argmin}_{s' \in \mathcal{M}_i} F(\alpha) + \langle \nabla_i F(\alpha), s' - \alpha_i \rangle.$$

$$\alpha_i = \alpha_i - \gamma(s - \alpha_i).$$

  - Can implement algorithm in terms of primal variables.

Block Coordinate Frank Wolfe

Key ideas behind BCFW for SSVMs:

- Dual problem has as the form

$$\min_{\alpha_i \in \mathcal{M}_i} F(\alpha) = f(A\alpha) - \sum_i f_i(\alpha_i).$$

  where f is smooth.

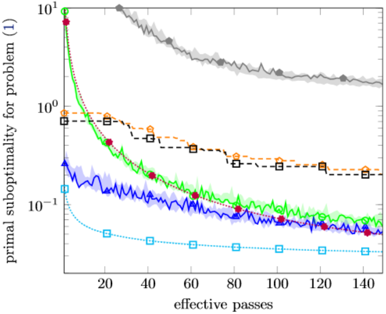
- Problem structure where we can use block coordinate descent:
  - Normal coordinate updates intractable because $\alpha_i \in |\mathcal{Y}|$.
  - But Frank-Wolfe block-coordinate update is equivalent to decoding

$$s = \underset{s' \in \mathcal{M}_i}{\text{argmin}}\, F(\alpha) + \langle \nabla_i F(\alpha), s' - \alpha_i \rangle.$$

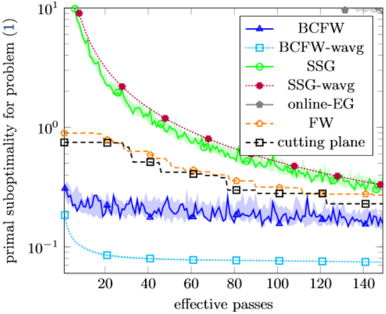
$$\alpha_i = \alpha_i - \gamma(s - \alpha_i).$$

  - Can implement algorithm in terms of primal variables.
- Connections between Frank-Wolfe and other algorithms:
  - Frank-Wolfe on dual problem is subgradient step on primal.
  - 'Fully corrective' Frank-Wolfe is equivalent to cutting plane.
  - Herding is a special case of Frank-Wolfe.

Comparison of SSVM Solvers



(b) OCR dataset, $\lambda = 0.001$.



(c) OCR dataset, $\lambda = 1/n$.

PGM Crash Course Summary

Week 1 covered exact inference:

1. Inference tasks, exact inference by enumeration.
2. Exact inference in chains and trees.
3. Conditional UGMs, exact inference with small cutsets.
4. Super-nodes and junction tree for inference in general graphs.
5. Exact Inference in semi-Markov chains and exact decoding in sub-modular models.

PGM Crash Course Summary

Week 2 started learning and approximate Inference:

1. Log-linear parameterization and parameter tieing, maximum likelihood in MRFs and CRFs.
2. Approximate decoding with ICM and alpha-expansions.
3. Approximate sampling with MCMC and herding.
4. Hidden variables, RBMs, and learning with Younes' algorithm.
5. Learning the graph structure.

PGM Crash Course Summary

Week 3 covered learning and inference with convex analysis:

1. Variational form of $\log(Z)$, maximum entropy, mean field.
2. Loopy belief propagation, Bethe and Kikiuchi free energies.
3. Convex variational upper bounds.
4. Convex relaxations of decoding.
5. Support vector machines and dual coordinate ascent.

Futher Topics: Types of Graphs

Alternatives to undirected graphs:

- Directed acyclic graphs (DAG):
  - AKA Bayesian networks.
  - More complicated marginal/conditional independence properties.
  - Unconditional inference and sampling are easy.
  - Learning without hidden variables is easy.
  - Everything else is as hard as UGMs in general.

Futher Topics: Types of Graphs

Alternatives to undirected graphs:

- Directed acyclic graphs (DAG):
  - AKA Bayesian networks.
  - More complicated marginal/conditional independence properties.
  - Unconditional inference and sampling are easy.
  - Learning without hidden variables is easy.
  - Everything else is as hard as UGMs in general.
- Decomposable graphs:
  - Graph representable by UGMs and DAGs.
  - Closed-form parameter learning.
- Chain graphs:
  - A DAG structure on UGM super-nodes.

Futher Topics: Types of Graphs

Alternatives to undirected graphs:

- Directed acyclic graphs (DAG):
  - AKA Bayesian networks.
  - More complicated marginal/conditional independence properties.
  - Unconditional inference and sampling are easy.
  - Learning without hidden variables is easy.
  - Everything else is as hard as UGMs in general.
- Decomposable graphs:
  - Graph representable by UGMs and DAGs.
  - Closed-form parameter learning.
- Chain graphs:
  - A DAG structure on UGM super-nodes.
- Ancestral graphs:
  - Generalization of DAGs that is closed under marginalization.
- More fancy things like sum-product networks, probabilistic context-free grammars, and probabilistic relational models.

Further Topics: Continuous Variables

Handling continuous variables:

- Gaussian graphical models (GGM):
  - Assume variables follow a joint Gaussian.
  - Special case of pairwise UGM: precision matrix gives graph.
  - Inference is easier: determinant vs. permanent.

Further Topics: Continuous Variables

Handling continuous variables:

- Gaussian graphical models (GGM):
  - Assume variables follow a joint Gaussian.
  - Special case of pairwise UGM: precision matrix gives graph.
  - Inference is easier: determinant vs. permanent.
- Mixed graphical models:
  - Continuous and discrete random variables.
  - Typically, assume that $p(\text{continuous}|\text{discrete})$ is Gaussian.

Further Topics: Continuous Variables

Handling continuous variables:

- Gaussian graphical models (GGM):
  - Assume variables follow a joint Gaussian.
  - Special case of pairwise UGM: precision matrix gives graph.
  - Inference is easier: determinant vs. permanent.
- Mixed graphical models:
  - Continuous and discrete random variables.
  - Typically, assume that $p(\text{continuous}|\text{discrete})$ is Gaussian.
- Copula models:
  - Instead of modeling joint PDF, model joint CDF.
- Transformed Gaussian models like the nonparamnormal.

Further Topics: Continuous Variables

Handling continuous variables:

- Gaussian graphical models (GGM):
  - Assume variables follow a joint Gaussian.
  - Special case of pairwise UGM: precision matrix gives graph.
  - Inference is easier: determinant vs. permanent.
- Mixed graphical models:
  - Continuous and discrete random variables.
  - Typically, assume that $p(\text{continuous}|\text{discrete})$ is Gaussian.
- Copula models:
  - Instead of modeling joint PDF, model joint CDF.
- Transformed Gaussian models like the nonparamnormal.
- Or just discretize!

Further Topics: Approximate Inference

Other approaches to inference:

- Exact inference possible in certain planar graphs.
- Fully-polynomial randomized approximation scheme (FPRAS):
  - Approximations based on fast-mixing Markov chains.
- Particle filters / Sequential Monte Carlo (SMC):
  - Useful for non-Gaussian time-series models.
- Expectation propagation (EP):
  - Loopy BP using moments from simpler model (e.g., Gaussian).

Futher Topics: Parameter Learning

Other training schemes:

- Primal-dual methods:
  - Alternate between updating primal and dual variables.
- Semi-supervised.
  - Learning when some examples have no labels.
- Marginal loss.
  - Maximize product of marginals (non-convex).
  - A better loss when interested in clasification errors.
- Posterior regularization.
  - Put regularizer posterior marginals.
  - E.g., each sentence can have one verb.

Further Topics: Feature Learning

Learning features and latent dynamics:

- Conditional neural fields (CNFs):
  - CRF model with node potentials from deep neural network.
- Latent-dynamic conditional random fields (LDCRFs):
  - CRF with latent chain-structure between features and labels.